NIST Special Publication 800-162

Guide to Attribute Based Access Control (ABAC) Definition and Considerations

Vincent C. Hu
David Ferraiolo
Rick Kuhn
Adam Schnitzer
Kenneth Sandlin
Robert Miller
Karen Scarfone

C O M P U T E R S E C U R I T Y

National Institute of Standards and Technology
U.S. Department of Commerce

NIST Special Publication 800-162

Guide to Attribute Based Access Control (ABAC) Definition and Considerations

Vincent C. Hu
David Ferraiolo
Rick Kuhn
Computer Security Division
Information Technology Laboratory

Adam Schnitzer
Booz Allen Hamilton
McLean, VA

Kenneth Sandlin
Robert Miller
The MITRE Corporation
McLean, VA

Karen Scarfone
Scarfone Cybersecurity
Clifton, VA

January 2014

U.S. Department of Commerce
Penny Pritzker, Secretary

National Institute of Standards and Technology
Patrick D. Gallagher, Under Secretary of Commerce for Standards and Technology and Director

Authority

This publication has been developed by NIST to further its statutory responsibilities under the Federal Information Security Management Act (FISMA), Public Law (P.L.) 107-347. NIST is responsible for developing information security standards and guidelines, including minimum requirements for Federal information systems, but such standards and guidelines shall not apply to national security systems without the express approval of appropriate Federal officials exercising policy authority over such systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), *Securing Agency Information Systems*, as analyzed in Circular A-130, Appendix IV: *Analysis of Key Sections*. Supplemental information is provided in Circular A-130, Appendix III, *Security of Federal Automated Information Resources*.

Nothing in this publication should be taken to contradict the standards and guidelines made mandatory and binding on Federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other Federal official. This publication may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright in the United States. Attribution would, however, be appreciated by NIST.

National Institute of Standards and Technology Special Publication 800-162
Natl. Inst. Stand. Technol. Spec. Publ. 800-162, 45 pages (January 2014)
CODEN: NSPUE2

Comments on this publication may be submitted to:

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in Federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Abstract

This document provides Federal agencies with a definition of attribute based access control (ABAC). ABAC is a logical access control methodology where authorization to perform a set of operations is determined by evaluating attributes associated with the subject, object, requested operations, and, in some cases, environment conditions against policy, rules, or relationships that describe the allowable operations for a given set of attributes. This document also provides considerations for using ABAC to improve information sharing within organizations and between organizations while maintaining control of that information.

Keywords

access control; access control mechanism; access control model; access control policy; attribute based access control (ABAC); authorization; privilege

Acknowledgements

The authors, Vincent C. Hu, David Ferraiolo, and Rick Kuhn of the National Institute of Standards and Technology (NIST); Adam Schnitzer of Booz Allen Hamilton; Kenneth Sandlin and Robert Miller of The MITRE Corporation; and Karen Scarfone of Scarfone Cybersecurity, wish to thank their colleagues who reviewed drafts of this document, including the following: Arthur R. Friedman, Alan J. Lang, Margaret M. Cogdell, and Kevin Miller from the National Security Agency (NSA), Jeffery L. Coleman (SOTERA Defense Solutions), Anne P. Townsend (The MITRE Corporation), Jennifer Newcomb (Booz Allen Hamilton), Tim Weil (Coalfire), Ed Coyne (DRC), John W. Tolbert (Boeing), Jeremy Wyant (General Dynamics), Ian Glazer (Gartner), Scott C. Fitch (Lockheed Martin), Tim Schmoyer (Jericho Systems), Luigi Logrippo (Université du Québec en Outaouais), Dave Coxe (Criterion Systems), Don Graham (Radiant Logic), and Ronald Ross, and Ramaswamy Chandramouli (NIST). Additionally, the NIST Computer Security Division would like to thank Mr. Friedman for initiating this effort and having the foresight to anticipate the growing importance of Attribute Based Access Control in government and industry.

The authors also gratefully acknowledge and appreciate the comments and contributions made by government agencies, private organizations, and individuals in providing direction and assistance in the development of this document.

Trademark Information

Table of Contents

List of Figures

Executive Summary

The concept of Attribute Based Access Control (ABAC) has existed for many years. It represents a point in the space of logical access control that includes access control lists, role-based access control, and the ABAC method for providing access based on the evaluation of attributes. Traditionally, access control has been based on the identity of a user requesting execution of a capability to perform an operation (e.g., read) on an object (e.g., a file), either directly, or through predefined attribute types such as roles or groups assigned to that user. Practitioners have noted that this approach to access control is often cumbersome to manage given the need to associate capabilities directly to users or their roles or groups. It has also been noted that the requester qualifiers of identity, groups, and roles are often insufficient in the expression of real-world access control policies. An alternative is to grant or deny user requests based on arbitrary attributes of the user and arbitrary attributes of the object, and environment conditions that may be globally recognized and more relevant to the policies at hand. This approach is often referred to as ABAC.

In November 2009, the Federal Chief Information Officers Council (Federal CIO Council) published the Federal Identity, Credential, and Access Management (FICAM) Roadmap and Implementation Plan v1.0 [FEDCIO1], which provided guidance to federal organizations to evolve their logical access control architectures to include the evaluation of attributes as a way to enable access within and between organizations across the Federal enterprise. In December 2011, the FICAM Roadmap and Implementation Plan v2.0 [FEDCIO2] took the next step of calling out ABAC as a recommended access control model for promoting information sharing between diverse and disparate organizations. In December 2012, the National Strategy for Information Sharing and Safeguarding included a Priority Objective that the Federal Government should extend and implement the FICAM Roadmap across Federal networks in all security domains. The U.S. General Services Administration (GSA) and the Federal CIO Council are designated leads for this Objective, and are preparing an implementation plan.

Despite the clear guidance to implement the FICAM Roadmap and contextual (risk adaptive) role or attribute based access control, to date there has not been a comprehensive effort to formally define or guide the implementation of ABAC within the Federal Government. This document serves a two-fold purpose. First, it aims to provide Federal agencies with a definition of ABAC and a description of the functional components of ABAC. Second, it provides planning, design, implementation, and operational considerations for employing ABAC within an enterprise with the goal of improving information sharing while maintaining control of that information. This document should not be interpreted as an analysis of alternatives between ABAC and other access-control capabilities, as it focuses on the challenges of implementing ABAC rather than on balancing the cost and effectiveness of other capabilities versus ABAC.

ABAC is a logical access control model that is distinguishable because it controls access to objects by evaluating rules against the attributes of entities (subject and object), operations, and the environment relevant to a request. ABAC systems are capable of enforcing both Discretionary Access Control (DAC) and Mandatory Access Control (MAC) concepts. ABAC enables precise access control, which allows for a higher number of discrete inputs into an access control decision, providing a bigger set of possible combinations of those variables to reflect a larger and more definitive set of possible rules to express policies.

The access control policies that can be implemented in ABAC are limited only by the computational language and the richness of the available attributes. This flexibility enables the greatest breadth of subjects to access the greatest breadth of objects without specifying individual relationships between each subject and each object. For example, a subject is assigned a set of subject attributes upon employment

(e.g., Nancy Smith is a *Nurse Practitioner* in the *Cardiology Department*). An object is assigned its object attributes upon creation (e.g., a folder with *Medical Records* of *Heart Patients)*. Objects may receive their attributes either directly from the creator or as a result of automated scanning tools. The administrator or owner of an object creates an access control rule using attributes of subjects and objects to govern the set of allowable capabilities (e.g., all *Nurse Practitioners* in the *Cardiology Department* can *View* the *Medical Records* of *Heart Patients).* Under ABAC, access decisions can change between requests by simply changing attribute values, without the need to change the subject/object relationships defining underlying rule sets. This provides a more dynamic access control management capability and limits long-term maintenance requirements of object protections.

Further, ABAC enables object owners or administrators to apply access control policy without prior knowledge of the specific subject and for an unlimited number of subjects that might require access. As new subjects join the organization, rules and objects do not need to be modified. As long as the subject is assigned the attributes necessary for access to the required objects (e.g., all Nurse Practitioners in the Cardiology Department are assigned those attributes), no modifications to existing rules or object attributes are required. This benefit is often referred to as accommodating the external (unanticipated) user and is one of the primary benefits of employing ABAC.

When deployed across an enterprise for the purposes of increasing information sharing among diverse organizations, ABAC implementations can become complex—supported by the existence of an attribute management infrastructure, machine-enforceable policies, and an array of functions that support access decisions and policy enforcement.

In addition to the basic policy, attribute, and access control mechanism requirements, the enterprise must support management functions for enterprise policy development and distribution, enterprise identity and subject attributes, subject attribute sharing, enterprise object attributes, authentication, and access control mechanism deployment and distribution. The development and deployment of these capabilities requires the careful consideration of a number of factors that will influence the design, security, and interoperability of an enterprise ABAC solution. These factors can be summarized around a set of activities:

- Establish the Business Case for ABAC Implementation
- Understand the Operational Requirements and Overall Enterprise Architecture
- Establish or Refine Business Processes to Support ABAC
- Develop and Acquire an Interoperable Set of Capabilities
- Operate with Efficiency

The remainder of this document provides a more detailed explanation of ABAC concepts and considerations for employment of enterprise ABAC capabilities. This document serves as a first step to help planners, architects, managers, and implementers fulfill the information sharing and protection requirements of the U.S. Federal Government, through the employment of ABAC.

1. Introduction

1.1 Purpose and Scope

The purpose of this document is to provide Federal agencies with a definition of **Attribute Based Access Control** (ABAC) and provide considerations for using ABAC to improve information sharing while maintaining control of that information. This document describes the functional components of ABAC, as well as a set of issues for employing ABAC within a large enterprise without directly addressing authentication mechanisms or all aspects of Identity Management[1], thus assuming subjects are bound to trusted identities or identity providers. The focus is on core ABAC concepts without addressing in detail topics such as Attribute Engineering/Management, Integration with Identity Management, Federation, Situational Awareness (Real Time or Contextual) Mechanism, Policy Management, and Natural Language Policy translation to Digital Policy. The discussed considerations should not be deemed comprehensive. Before selecting and deploying an ABAC product or technology, the hosting organization should augment these considerations with testing and independent product reviews.

This document brings together many previously separate bodies of ABAC knowledge in order to bridge gaps between available technology and best practice ABAC implementations, and strives to provide guidelines that can be consistently applied throughout organizations. It can be used as an informational guide for organizations that are considering deploying, planning to deploy, or are currently deploying ABAC systems.

This guidance extends the information in NISTIR 7316, *Assessment of Access Control Systems* [NIST7316]; NISTIR 7665, *Proceedings of the Privilege Management Workshop* [NIST7665]; NISTIR 7657, *A Report on the Privilege (Access) Management Workshop* [NIST7657]; and NISTIR 7874, *Guidelines for Access Control System Evaluation Metrics* [NIST7874], which demonstrates the fundamental concepts of policy, models, and properties of Access Control (AC) systems.

1.2 Audience

This document is intended to benefit and address the needs of two specific audiences:

- Persons who have a basic understanding of access control concepts and desire a general understanding of ABAC concepts
- Access control subject matter experts or managers experienced in access control concepts who are seeking detailed deployment or operational information on ABAC

1.3 Document Structure

The rest of this document is divided into the following sections and appendixes:

- Section 2 provides a basic understanding of ABAC. It gives readers an overview of the current state of logical access control, a working definition of ABAC, and an explanation of core and enterprise level ABAC concepts. Readers can gain a general understanding of ABAC concepts from just completing Section 2.
- Section 3 discusses ABAC enterprise employment considerations during the initiation, acquisition/development, implementation/assessment, and operations/maintenance phases.

---

[1] See NIST SP 800-63-1 at http://csrc.nist.gov/publications/nistpubs/800-63-1/SP-800-63-1.pdf and NIST SP 800-63-2 at http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-2.pdf.

Readers with an interest in access control and/or project management will benefit most from this section.
- Section 4 concludes the document.
- Appendix A defines various acronyms and abbreviations related to ABAC.
- Appendix B lists the references for the document.

Because of the constantly changing nature of the IT industry, readers are strongly encouraged to take advantage of other resources, including those referenced in this document.

1.4 Notes on Terminology

The terminology is not meant to be mandatory, but is intended to be consistent within the confines of the document itself. Where possible, terminology that is used elsewhere within NIST publications and across the Federal Government was adopted to maintain consistency. Where terms were found to be used inconsistently or where multiple terms were being used throughout the Federal Government and the Identity and Access Control community to address a common concept, the most concise terms and definitions were applied.

A logical **object**—sometimes referred to as a **resource**—is an entity to be protected from unauthorized use. The **subject** represents the entity requesting to perform an operation upon the object. The subject is sometimes referred to as a **requestor**.

The subject is most often assumed to be a human. A **non-person entity (NPE)**, such as an autonomous service or application could also fill the role of the subject. In general, every operation performed by a computer must be done on behalf of some person or organization (in the case of an NPE) with the authority to perform the operation. The term **subject** is used to denote a human or NPE requesting access to an object.

There are characteristics or **attributes** of a subject such as name, date of birth, home address, training record, and job function that may, either individually or when combined, comprise a unique identity that distinguishes that person from all others. These characteristics are often called **subject attributes.** The term **subject attributes** is used consistently throughout this document.

In the course of a person's life, he or she may work for different organizations, may act in different roles, and may inherit different **privileges** tied to those roles. The person may establish different **personas** for each organization or role and amass different attributes related to each persona. For example, an individual may work for Company A as a gate guard during the week and may work for Company B as a shift manager on the weekend. The subject attributes are different for each persona. Although trained and qualified as a Gate Guard for Company A, while operating in her Company B persona as a shift manager on the weekend she does not have the authority to perform as a Gate Guard for Company B.

Authentication is not the same as access control or authorization. **Authentication** is the act of verifying that the subject has been authorized to use the presented identifier by a trusted identity provider organization. **Access control** or **authorization**, on the other hand, is the decision to permit or deny a subject access to system objects (network, data, application, service, etc.) Note that ABAC can be used without identification information, and authentication method is not addressed in this document. The terms **access control** and **authorization** are used synonymously throughout this document.

Privileges represent the authorized behavior of a subject; they are defined by an authority and embodied in **policy** or rules. For the purposes of this document, the terms **privileges** and **authorizations** are used

interchangeably in that they are meant to convey one's authority and implicit approval to access one or more objects.

Environment conditions are dynamic factors, independent of subject and object, that may be used as attributes at decision time to influence an access decision. Examples of environment conditions include time, location, threat level, and temperature.

Policy, rules, and relationships govern allowable behavior within an organization, based on the privileges of subjects and how resources or objects are to be protected under which environment conditions. Throughout this document, the term **policy** is used to convey these rules and relationships. Policy is typically written from the perspective of the object that needs protecting and the privileges available to subjects.

Like subjects, each object has a set of attributes that help describe and identify it. These traits are called **object attributes** and are sometimes referred to as **resource attributes**. This document uses the term **object attributes** consistently throughout. Object attributes are typically bound to their objects through reference, by embedding them within the object, or through some other means of assured association such as cryptographic binding.[2]

Information about policy, such as author, policy effective date, deconflict methods, etc. are sometimes called **metapolicy**. Information about attributes such as attribute authority, attribute creation date, etc. are sometimes called **metaattributes**. Metapolicy and metaattributes may be used in the development of sets of policies and the identification of the appropriate attribute sets needed for authorization. A good example of the use of a metaattribute is assigning an assurance level or measure of confidence to the attribute—a composite score for an attribute that could combine subjective ratings like a confidence score for the authority behind the attribute, a freshness score of the information in the attribute, and a level of accuracy score for how often the information is validated. At times, these measures of confidence may even be used as input to the access decision.

These policies must be enforced through some type of **access control mechanism**. The access control mechanism must assemble authorization information, which may include information about the object being protected, the subject requesting access, the policies governing access to the resource, and any contextual information needed to make a decision. By evaluating each policy element against the available information, the access control mechanism often employs a **policy decision point (PDP)** to render a decision, a **policy enforcement point (PEP)** to enforce the decision, and some sort of **context handler** or **workflow coordinator** to manage the collection of attributes required for the decision. For the purposes of this document, it is assumed that the term **access control mechanism** incorporates all of this functionality, and the term is used throughout.

[2] Cryptographic binding is a methodology for providing integrity and authenticity to data and data relationships using well-known cryptographic techniques. Cryptographic binding works by determining the hash value of each object attribute associated with a specific object and digitally signing the collection of hashed values. When the object is accessed, if the object signature fails, the attribute hash values are then compared to determine which element was modified since the last binding operation.

2. Understanding ABAC

Fully understanding ABAC requires understanding of the basic principles of logical access control. The purpose of logical access control is to protect objects—be they data, services, executable applications, network devices, or some other type of information technology—from unauthorized operations. These operations may include discovering, reading, creating, editing, deleting, and executing objects. These objects are owned by an individual or organization and have some inherent value that motivates those owners to protect them. As owners of the objects, they have the authority to establish a policy that describes what operations may be performed upon those objects, by whom, and in what context those subjects may perform those operations. If the subject satisfies the access control policy established by the object owner, then the subject is authorized to perform the desired operation on that object—better known as being granted access to the object. If the subject does not satisfy the policy, then it is denied access to the object.

Computer security architects and administrators deploy access control mechanisms (ACM) in logic aligned to protect their objects by mediating requests from subjects. These ACMs can use a variety of methods to enforce the access control policy that applies to those objects. The ACM can be defined as:

> ***Access Control Mechanism (ACM):*** *The logical component that serves to receive the access request from the subject, to decide, and to enforce the access decision.*

How these ACMs function can be described in terms of various logical access control models. These access control models provide a framework and set of boundary conditions upon which the objects, subjects, operations, and rules may be combined to generate and enforce an access control decision. Each model has its own advantages and limitations but it is important to note the evolution of these models to fully appreciate the flexibility and applicability of the ABAC model.

MAC/DAC

An early application of logical access control was applied in Department of Defense (DoD) applications in the 1960s and 1970s with the emergence of the concepts of Discretionary Access Control (DAC) and Mandatory Access Control (MAC). These terms are further defined in the DoD Trusted Computer System Evaluation Criteria (TCSEC) or "Orange Book" [TCSEC]. The definition of DAC and MAC can be also found in [NIST800-53].

IBAC/ACLs

As networks grew, the need to limit access to specific protected objects spurred the growth of identity based access control (IBAC) capabilities. IBAC employs mechanisms such as access control lists (ACLs) to capture the identities of those allowed to access the object. If a subject presents a credential that matches the one held in the ACL, the subject is given access to the object. Individual privileges of the subject to perform operations (read, write, edit, delete, etc.) are managed on an individual basis by the object owner. Each object needs its own ACL and set of privileges assigned to each subject. In the IBAC model, the authorization decisions are made prior to any specific access request and result in the subject being added to the ACL. For each subject to be placed on an ACL, the object owner must evaluate identity, object, and context attributes against policy governing the object and decide whether to add the subject to the ACL. This decision is static and a notification process is required for the owner to reevaluate and perhaps remove a subject from the ACL to represent subject, object, or contextual changes. Failure to remove or revoke access over time leads to users accumulating privileges.

RBAC
Role-Based Access Control model (RBAC) [FK92, ANSI349, Sandhu96] employs pre-defined roles that carry a specific set of privileges associated with them and to which subjects are assigned. For example, a subject assigned the role of Manager will have access to a different set of objects than someone assigned the role of Analyst. In this model, access is implicitly predetermined by the person assigning the roles to each individual and explicitly by the object owner when determining the privilege associated with each role. At the point of an access request, the access control mechanism evaluates the role assigned to the subject requesting access and the set of operations this role is authorized to perform on the object before rendering and enforcing an access decision. Note that a role may be viewed as a subject attribute that is evaluated by the access control mechanism and around which object access policy is generated. As the RBAC specification gained popularity, it made central management of enterprise access control capabilities possible and reduced the need for ACLs.

ABAC
ACLs and RBAC are in some ways special cases of ABAC in terms of the attributes used. ACLs work on the attribute of "identity". RBAC works on the attribute of "role". The key difference with ABAC is the concept of policies that express a complex Boolean rule set that can evaluate many different attributes. While it is possible to achieve ABAC objectives using ACLs or RBAC, demonstrating AC requirements compliance is difficult and costly due to the level of abstraction required between the AC requirements and the ACL or RBAC model. Another problem with ACL or RBAC models is that if the AC requirement is changed, it may be difficult to identify all the places where the ACL or RBAC implementation needs to be updated.

One example of an access control framework that is consistent with ABAC is the Extensible Access Control Markup Language (XACML) [XACML]. The XACML model employs elements such as rules, policies, rule- and policy-combining algorithms, attributes (subject, (resource) object, action and environment conditions), obligations, and advice. Its reference architecture includes functions such as Policy Decision Points (PDPs), Policy Enforcement Points (PEPs), Policy Administration Points (PAPs), and Policy Information Points (PIPs) to control access. Another example is the Next Generation Access Control standard [ANSI499].

In general, ABAC avoids the need for capabilities (operation/object pairs) to be directly assigned to subject requesters or to their roles or groups before the request is made. Instead, when a subject requests access, the ABAC engine can make an access control decision based on the assigned attributes of the requester, the assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions. Under this arrangement policies can be created and managed without direct reference to potentially numerous users and objects, and users and objects can be provisioned without reference to policy.

2.1 The Benefit of ABAC

In many AC systems, logical access control solutions have been based primarily on the identity of a subject requesting execution of an operation (e.g., read) upon an object (e.g., a file). Examples include IBAC or RBAC where access to an object has been individually granted to a locally identified subject, or when access to an object has been granted to locally defined roles that the subject is a member of. This approach to AC is often cumbersome to manage. In this non-ABAC multi-organizational access method example (illustrated below in Figure 1), authenticated access to objects outside of the subject's originating organization would require the subject's identity to be pre-provisioned in the target organization and pre-populated on an access list.

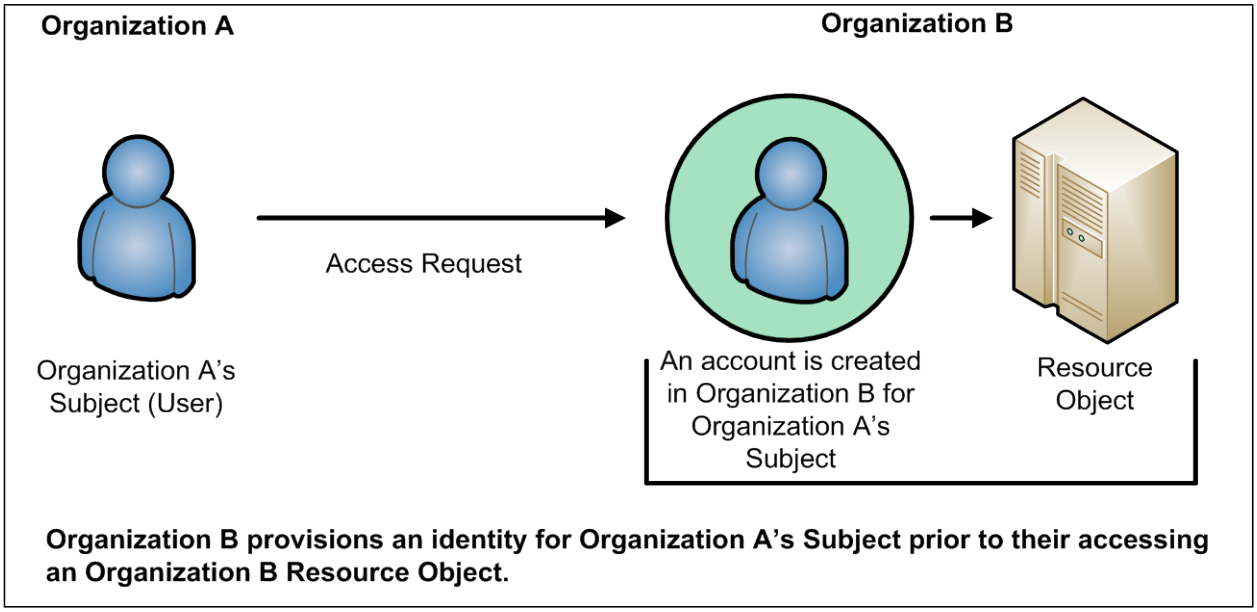


Figure 1: Traditional (Non-ABAC) Multi-Organizational Access Method

Additionally, the subject qualifiers, such as identity and roles, are often insufficient in the expression of real-world AC needs. RBAC makes a decision based on the subject's association with a role. RBAC does not easily support multi-factor decisions (for example, decisions dependent on physical location, and specialized training such as for Health Insurance Portability and Accountability Act (HIPAA) records access; recent training on HIPAA data protection may be a prerequisite to view medical records.) RBAC role assignments tend to be based upon more static organizational positions, presenting challenges in certain RBAC architectures where dynamic access control decisions are required. Trying to implement these kinds of access control decisions would require the creation of numerous roles that are ad hoc and limited in membership, leading to what is often termed "role explosion".

A method is needed to make AC decisions without previous knowledge of the object by the subject or knowledge of the subject by the object-owner. By relying upon the concepts of subject and object attributes consistently defined between organizations, ABAC avoids the need for explicit authorizations to be directly assigned to individual subjects prior to a request to perform an operation on the object. Moreover, this model enables flexibility in a large enterprise where management of access control lists or roles and groups would be time consuming and complex. Leveraging consistently defined attributes that span both subjects and objects, authentication and authorization activities can be executed and administered in the same or separate infrastructures, while maintaining appropriate levels of security.

2.2 A Working Definition of ABAC

ABAC has been described in various ways. For example, one early paper on web services states that ABAC "grants accesses to services based on the attributes possessed by the requester" [WWJ04], while a discussion of security in geographic information systems describes ABAC as an approach in which "attribute values associated with users determine the association of users with privileges" [CGLO09].

Still another paper summarizes ABAC as a model that is "based on subject, object, and environment attributes and supports both mandatory and discretionary access control needs" [YT05]. In these and other definitions, there is a reasonable consensus that ABAC determines access (i.e., operations upon system

objects) by matching the current value of subject attributes, object attributes, and environment conditions with the requirements specified in access control rules. The following is a high-level definition of ABAC:

Attribute Based Access Control (ABAC): *An access control method where subject requests to perform operations on objects are granted or denied based on assigned attributes of the subject, assigned attributes of the object, environment conditions, and a set of policies that are specified in terms of those attributes and conditions.*

Attributes are characteristics of the subject, object, or environment conditions. Attributes contain information given by a name-value pair.

A **subject** is a human user or NPE, such as a device that issues access requests to perform operations on objects. Subjects are assigned one or more attributes. For the purpose of this document, assume that subject and user are synonymous.

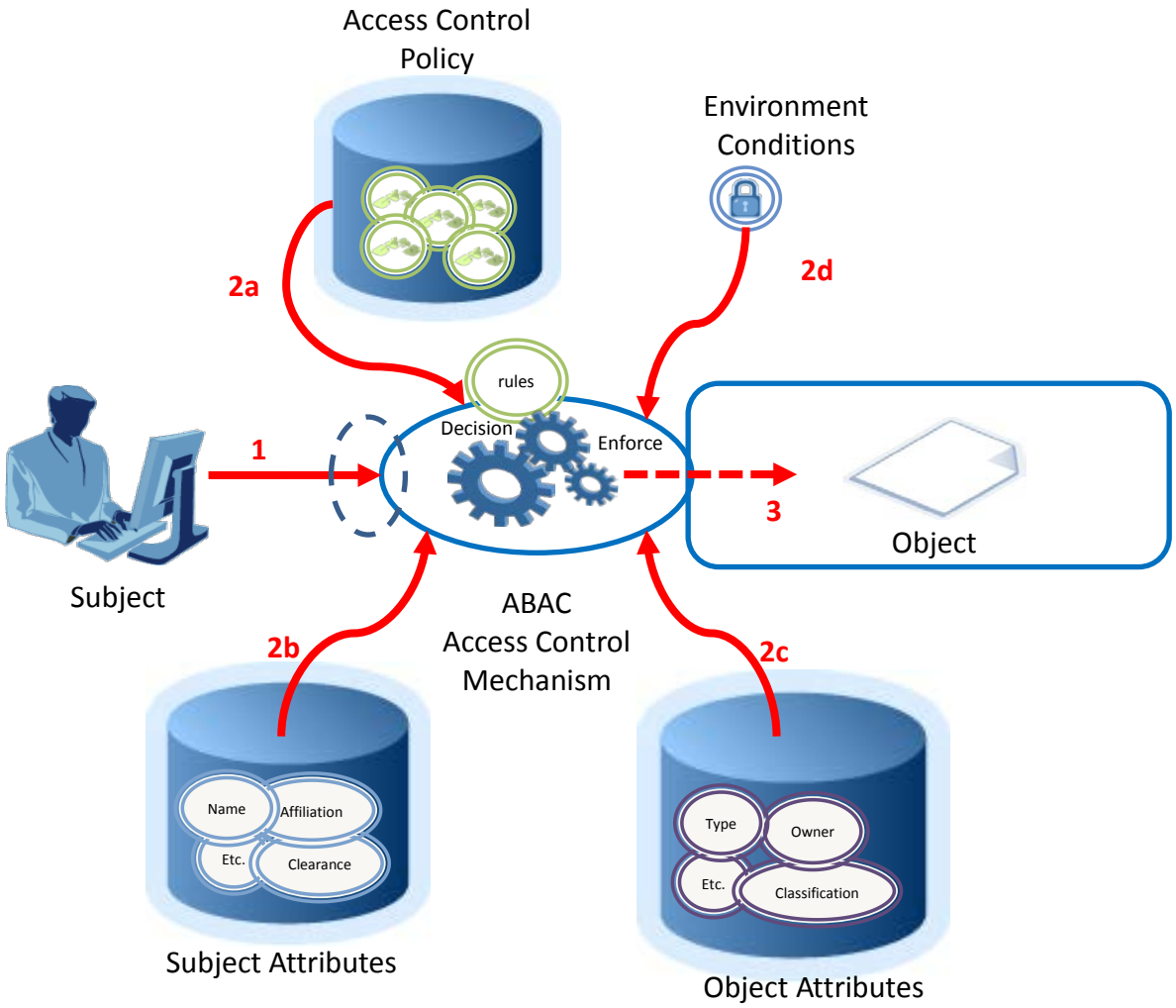
An **object** is a system resource for which access is managed by the ABAC system, such as devices, files, records, tables, processes, programs, networks, or domains containing or receiving information. It can be the resource or requested entity, as well as anything upon which an operation may be performed by a subject including data, applications, services, devices, and networks.

An **operation** is the execution of a function at the request of a subject upon an object. Operations include read, write, edit, delete, copy, execute, and modify.

Policy is the representation of **rules** or **relationships** that makes it possible to determine if a requested access should be allowed, given the values of the attributes of the subject, object, and possibly environment conditions.

Environment conditions: operational or situational context in which access requests occur. Environment conditions are detectable environmental characteristics. Environment characteristics are independent of subject or object, and may include the current time, day of the week, location of a user, or the current threat level.

The high-level ABAC definition is depicted in Figure 2 where the ABAC ACM receives the subject's access request, then examines the subject's and object's attributes against a specific policy. The ACM then determines what operations the subject may perform upon the object.



1. Subject requests access to object
2. Access Control Mechanism evaluates a) Rules, b) Subject Attributes, c) Object Attributes, and d) Environment Conditions to compute a decision
3. Subject is given access to object if authorized

Figure 2: Basic ABAC Scenario

2.3 Basic ABAC Concepts

In its most basic form, ABAC relies upon the evaluation of attributes of the subject, attributes of the object, environment conditions, and the formal relationship or access control rule or policy defining the allowable operations for subject-object attribute combinations. All ABAC solutions contain these basic

core capabilities to evaluate attributes and enforce rules or relationships between those attributes (see Figure 3 below).

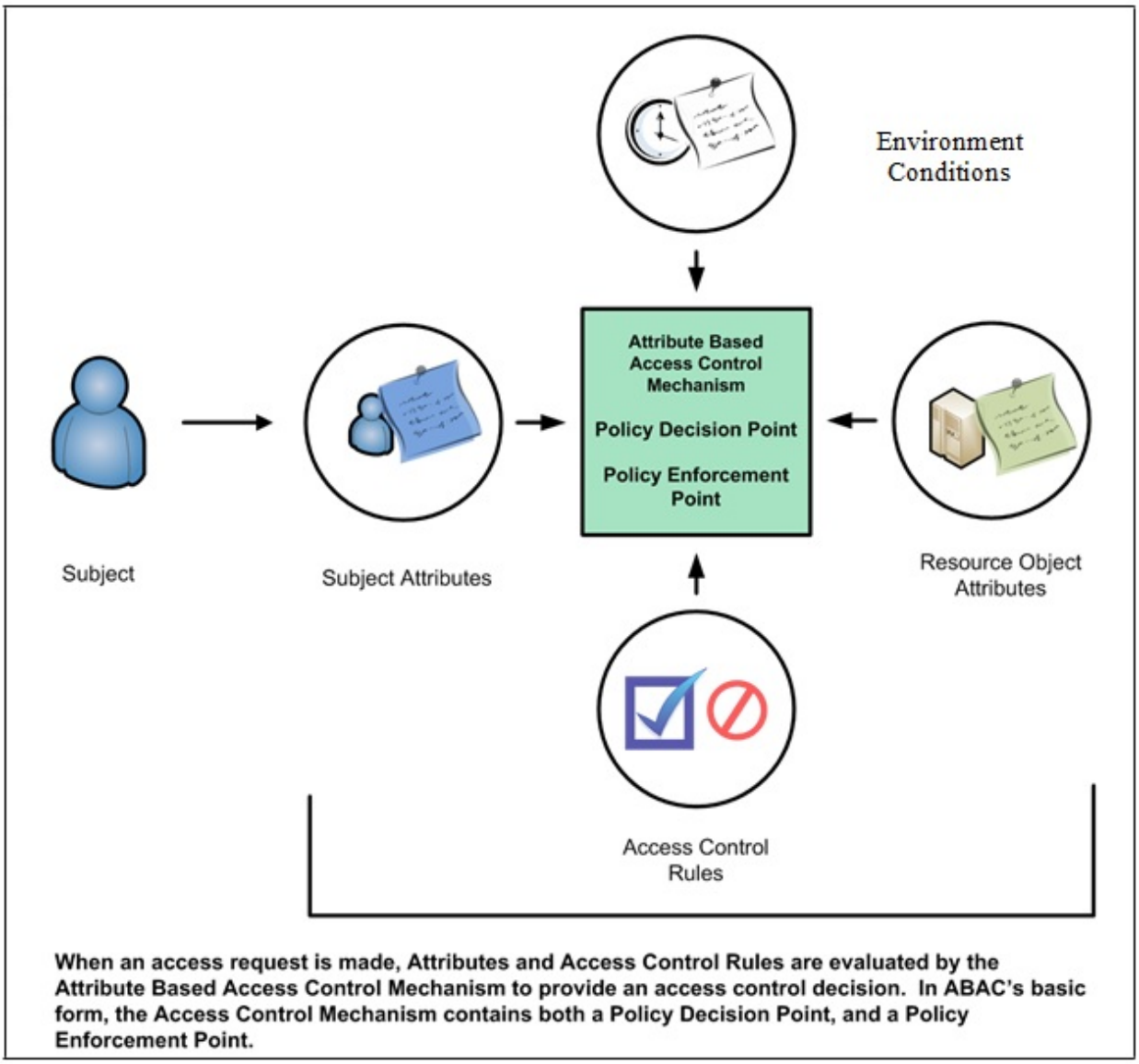


Figure 3: Core ABAC Mechanisms

Even within a small isolated system, ABAC relies upon the assignment of attributes to subjects and objects, and the development of policy that contains the access rules. Each object within the system must be assigned specific object attributes that characterize the object. Some attributes pertain to the entire instance of an object, such as the owner. Other attributes may only apply to parts of the object. For example, a document object could be owned by organization A, have a section with intellectual property from organization B, and be part of a program run by organization C. As another example, consider a document residing in a directory within a file management system. This document has a title, an author, a date of creation, and a date of last edit—all object attributes that are determined by the creator, author, or editor of the document. Additional object attributes may be assigned such as owning organization, intellectual property characteristics, export control classification, or security classification. Each time a

new document is created or modified, these object attributes must be captured. These object attributes are often embedded within the document itself, but they may be captured in a separate table, incorporated by reference, or managed by a separate application.

Each subject that uses the system must be assigned specific attributes. Consider the example of a user accessing a file management system. The user is established as a subject within the system by an administrator and characteristics about that user are captured as subject attributes. This subject may have a name, a role, and an organization affiliation. Other subject attributes may include US Person status, nationality, and security clearance. These subject attributes are assigned and managed by an authority within the organization that maintains the subject identity information for the file management system. As new users arrive, old users leave, and characteristics of subjects change, these subject attributes may need to be updated.

Every object within the system must have at least one policy that defines the access rules for the allowable subjects, operations, and environment conditions to the object. This policy is normally derived from documented or procedural rules that describe the business processes and allowable actions within the organization. For example, in a hospital setting, a rule may state that only authorized medical personnel shall be able to access a patient's medical record. In some system, if the object is a document with a *Record Type Attribute* of *Patient Medical Record*, then the *Medical Record Rule* will be selected and processed so that the subject with a *Personnel Type Attribute* value of *Non-Medical Support Staff* trying to perform the *Read* operation will be denied access and the operation will be disallowed. Note that this is only one approach to implementing the connection between attributes and rules.

The rules that bind subject and object attributes indirectly specify privileges (i.e., which subjects can perform which operations on which objects). Allowable operation rules can be expressed through many forms of computational language such as:

- A Boolean combination of attributes and conditions that satisfy the authorization for a specific operation
- A set of relations associating subject and object attributes and allowable operations

Once object attributes, subject attributes, and policies are established, objects can be protected using ABAC. Access control mechanisms mediate access to the objects by limiting access to allowable operations by allowable subjects. The ACM assembles the policy, subject attributes, and object attributes, then renders and enforces a decision based on the logic provided in the policy. ACMs must be able to manage the process required to make and enforce the decision, including determining what policy to retrieve, which attributes to retrieve in what order, and where to retrieve attributes. The ACM must then perform the computation necessary to render a decision.

The policies that can be implemented in an ABAC model are limited only to the degree imposed by the computational language and the richness of the available attributes. This flexibility enables the greatest breadth of subjects to access the greatest breadth of objects without having to specify individual relationships between each subject and each object. For example, a subject is assigned a set of subject attributes upon employment (e.g., Nancy Smith is a *Nurse Practitioner* in the *Cardiology Department*). An object is assigned its object attributes upon creation (e.g., a folder with *Medical Records* of *Heart Patients).* A designated authority creates rules to govern the set of allowable operations (e.g., all *Nurse Practitioners* in the *Cardiology Department* can *View* the *Medical Records* of *Heart Patients).* Adding to the flexibility, attributes and their values may then be modified throughout the lifecycle of subjects, objects, and attributes.

Provisioning attributes to subjects and objects governed by a ruleset that specifies what operations can take place enables an unlimited number of subjects to perform operations on the object—all without prior knowledge of the specific subject by the object-owner or rule-maker. As new subjects join the organization, rules and objects do not need to be modified. As long as the subject is assigned the attributes necessary for access to the required objects (e.g., all Nurse Practioners in the Cardiology Department are assigned those attributes), no modifications to existing rules or object attributes are required. This benefit is often referred to as accommodating the external (unexpected) user and is one of the primary benefits of employing ABAC.

Contrary to some other schemes, under the definition of ABAC presented here, operations do not have "attributes". As defined "Attributes contain information given by a name-value pair". For example, "read = all" (or "all = read") is not appropriate. Operations can have many types or classes, which are not "attributes" but a fixed set of values. It would be possible to make operation itself an "attribute name", such as "operation = read", but this would then be the only attribute for operation, which would be redundant.

To meet accountability requirements, there will be a need to track accesses of objects to specific subjects linked to specific users. Accountability could be lost if access decisions are based on attributes, but subject or user IDs are not tracked to specific access requests and decisions.

2.4 Enterprise ABAC Concepts

While ABAC is an enabler of information sharing, when deployed across an enterprise, the set of components required to implement ABAC gets more complex. At the enterprise level the increased scale requires complex and sometimes independently established management capabilities necessary to ensure consistent sharing and use of policies and attributes and the controlled distribution and employment of access control mechanisms throughout the enterprise. The following represents a definition of enterprise for this document.

Enterprise: *A collaboration or federation among entities for which information sharing is required and managed.*

Figure 4 below presents an example of the major components required to enable enterprise ABAC. Some enterprises have existing capabilities that can be leveraged to implement ABAC. For example, most enterprises have some form of identity and credential management to manage population of subject attributes, such as name, unique identifier, role, clearance, etc. Similarly, many enterprises may have some organizational policy or guidelines to establish rules authorizing subjects' access to enterprise objects. However, these rules are usually not written in a machine-enforceable format that can be integrated consistently across all applications. ABAC policies must be made available in machine-enforceable format, and stored in repositories and published for ACM consumption. These digital policies include subject and object attributes required to render access control decisions. The enterprise subject attributes must be created, stored, and shared across organizations within the enterprise through a subject attribute management capability. Likewise, enterprise object attributes must be established and bound to objects through an object attribute management capability. At this point, the ABAC-enabled access control mechanisms must be deployed. The remainder of this section provides more detail on each of these major components of enterprise ABAC.

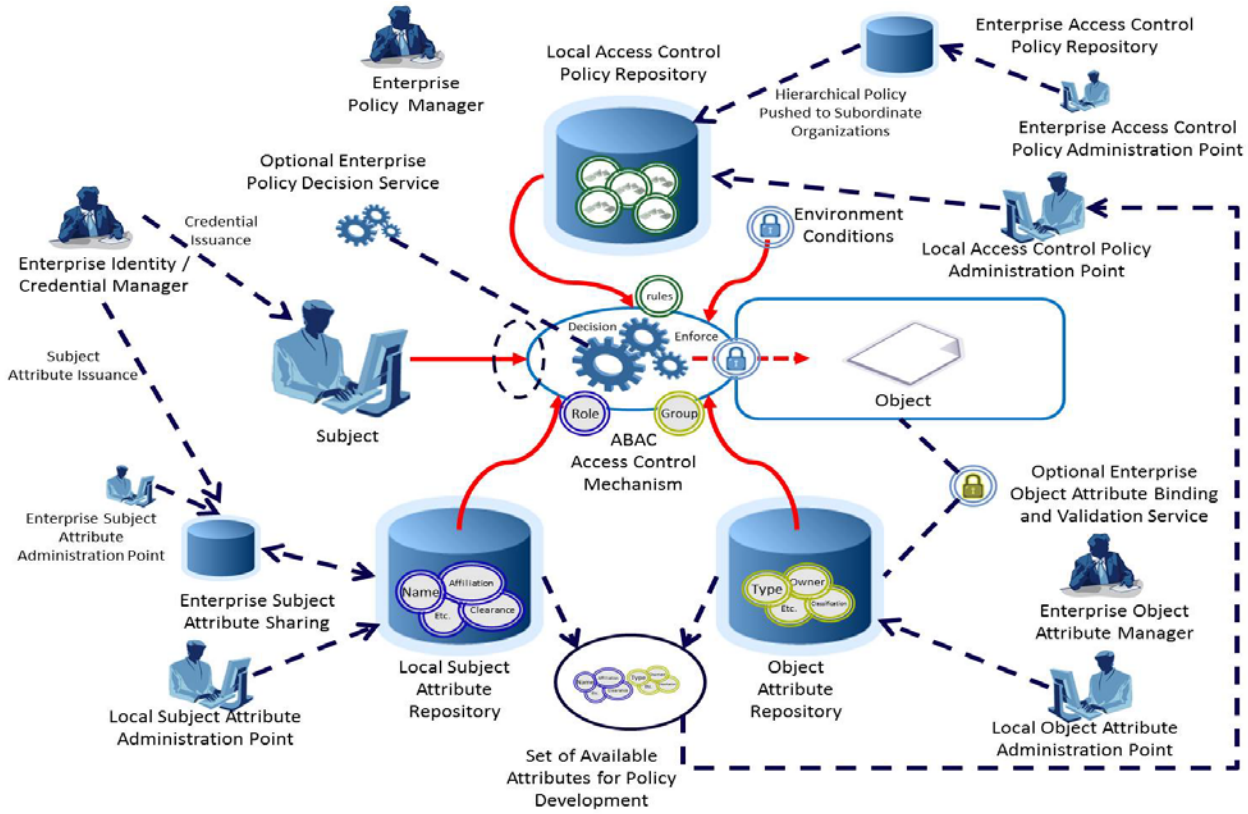


Figure 4: Enterprise ABAC Scenario Example

2.4.1 Enterprise ABAC Policy

Natural Language Policies (NLPs) are high-level requirements that specify how information access is managed and who, under what circumstances, may access what information. NLPs are expressed in human understandable terms and may not be directly implementable in an ACM. NLPs may be ambiguous and thus hard to derive in formally actionable elements, so the enterprise policy may be difficult to encode in machine-enforceable form. While NLPs can be application-specific and thus taken into consideration by the application system, NLPs are just as likely to pertain to subject actions that span multiple applications. For instance, NLPs may pertain to object usage within or across organizational units or may be based on need-to-know, competence, authority, obligation, or conflict-of-interest factors. Such policies may span multiple computing platforms and applications. NLPs are defined in this document as follows:

Natural Language Policy (NLP): *Statements governing management and access of enterprise objects. NLPs are human expressions that can be translated to machine-enforceable access control policies.*

Given that relevant NLPs exist for each organization in an enterprise, the next step is to translate those into a common set of rules that can be enforced equally and consistently within the ACMs across the enterprise. In order to accomplish this, it is necessary to identify all required subject/object attribute combinations and allowable operations. Often these values will vary from organization to organization and may require some form of consensus or mapping to each organization's existing attributes to accommodate enterprise interoperability. The agreed-upon list of subject and object attributes, the

allowable operations, and all mappings from existing organization-specific attributes are then translated into machine-enforceable format.

NLPs must be codified into Digital Policy (DP) algorithms or mechanisms. For efficiency of performance and simplicity in specification, an NLP may require decomposition and translation into different DPs that suit the infrastructure of operation units in the enterprise. DPs are defined in this document as:

Digital Policy (DP): *Access control rules that compile directly into machine executable codes or signals. Subject/object attributes, operations, and environment conditions are the fundamental elements of DP, the building blocks of DP rules, which are enforced by an access control mechanism.*

Multiple DPs may require Metapolicies (MPs), or policies dictating the use and management of DPs to handle DP hierarchical authorities, DP deconfliction, and DP storage and updates. MPs are used for managing DPs. Depending on the level of complexities, hierarchical MPs may be required based on the structures for the priority and combination strategies specified by NLP. MP is defined in this document as:

Metapolicy (MP): *A policy about policies, or policy for managing policies, such as assignment of priorities and resolution of conflicts between DPs or other MPs.*

Once DPs and MPs are developed they need to be managed, stored, validated, updated, prioritized, deconflicted, shared, retired, and enforced. Each of these operations requires a set of capabilities that will often be distributed across the enterprise and is collectively termed Digital Policy Management (DPM). There may be multiple policy authorities and hierarchies within organizations that have variations on enterprise policy. The rules for how DPs and MPs are managed may be determined by a central authority.

Proper DP definition and development are critical to the identification of subject and object attributes that are needed to render an access control decision. Remember that a DP statement is comprised of the subject and object attribute pairings as well as environment conditions needed to satisfy a set of allowable operations. Once the full set of subject and object attributes needed to satisfy the entire set of allowable operations for a given set of enterprise objects is identified, this set of attributes comprises the entire set of attributes needed to be defined, assigned, shared, and evaluated for enterprise ABAC access decisions. For this reason, identifying the NLP and DP must be accomplished by the support of attributes when implementing an enterprise ABAC capability. Additional considerations for management of DP can be found in Section 3 of this document.

2.4.2 Attribute Management in Enterprise ABAC

Next, consider the lists of attributes developed while examining the NLPs and DPs. Without a sufficient set of object and subject attributes, ABAC does not work. Attributes need to be named, defined, given a set of allowable values, assigned a schema, and associated to subjects and objects. Subject attributes need to be established, issued, stored, and managed under an authority. Object attributes must be assigned to the objects. Attributes shared across organizations should be located, retrieved, published, validated, updated, modified, and revoked.

Subject attributes are provisioned by attribute authorities—typically authoritative for the type of attribute that is provided and managed through an attribute administration point. Often, there are multiple authorities, each with authority over different attributes. For example, Security might be the authority for Clearance attributes, while Human Resources might be the authority for Name attributes. Subject attributes that need to be shared to allow subjects from one organization to access objects in another organization must be consistent, comparable, or mapped to allow equivalent policies to be enforced. For

example, a member of Organization A with the role Job Lead wants to access information in Organization B, except Organization B uses the term Task Lead to denote the equivalent role. This problem also applies to mapping between an enterprise attribute schema and an application-specific schema, particularly ones built before the enterprise schema is defined and/or Commercial Off-The-Shelf (COTS) products that come with their own built-in schema. Organizations must normalize subject attribute names and values, or maintain a mapping of equivalent terms for all organizations. This should be managed by a central authority.

Object attributes need to be established, maintained, and assigned to objects as objects are created or modified. While it may not be necessary to have a common set of object attributes in use across the enterprise, object attributes should be consistently employed to fulfill enterprise policy requirements, and available sets of object attributes should be published for those wishing to mark, tag, or otherwise apply object attributes to their objects. At times, it might be necessary to ensure that object attributes are not tampered with or altered to satisfy an access request. Objects can be cryptographically bound to their object attributes to identify whether objects or their corresponding attributes have been inappropriately modified. Mechanisms must be deployed to ensure that all objects created are assigned the appropriate set of object attributes to satisfy the policy being employed by the ACM. It may be necessary to have an Enterprise Object Attribute Manager to coordinate these requirements.

In the course of managing attributes, the concept of "metaattributes"—or characteristics of attributes—arises. Metaattributes apply to subjects, objects, and environment conditions as extended attribute information useful for enforcing more detailed policy that incorporates information about the attributes and for managing the volumes of data needed for enterprise attribute management. Metaattributes are defined in this document as:

Metaattributes: *Information about attributes necessary to implement MP and DP processing within an ACM.*

Additional considerations for attribute management can be found in Section 3 of this document.

2.4.3 Access Control Mechanism Distribution in Enterprise ABAC

Finally, consider the distribution and management of ACMs throughout the enterprise. Depending on the needs of the users, size of the enterprise, distribution of the resources, and sensitivity of the objects that need to be accessed or shared, the distribution of ACMs can be critical to the success of an ABAC implementation. The functional components of an ACM may be physically and logically separated and distributed within an enterprise rather than centralized as described in the system-level view of ABAC.

Within the ACM are several functional "points" that are the service node for retrieval and management of the policy, along with some logical components for handling the context or workflow of policy and attribute retrieval and assessment. Figure 5 shows the main functional points: the Policy Enforcement Point (PEP), the Policy Decision Point (PDP), the Policy Information Point (PIP), and the Policy Administration Point (PAP). When these components are in an environment, they must function together to provide access control decisions and policy enforcement.

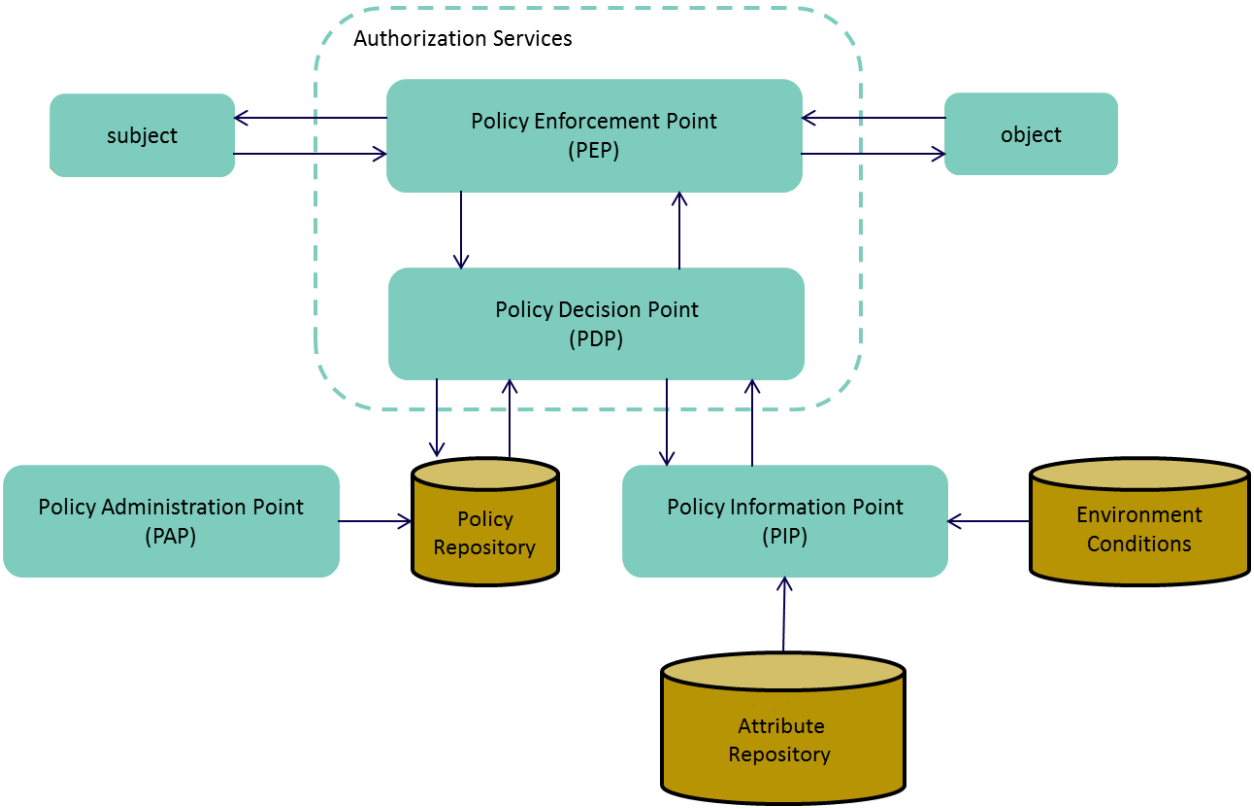


Figure 5: An Example of ACM Functional Points

A PDP performs an evaluation on DPs and MPs in order to produce an access control decision. PDP and PEP are defined in this document as follows:

Policy Decision Point (PDP): *Computes access decisions by evaluating the applicable DPs and MPs. One of the main functions of the PDP is to mediate or deconflict DPs according to MPs.*

The PEP enforces decisions made by the PDP:

Policy Enforcement Point (PEP): *Enforces policy decisions in response to a request from a subject requesting access to a protected object; the access control decisions are made by the PDP.*

PDP and PEP functionality can be distributed or centralized, and may be physically and logically separated from each other. For example, an enterprise could establish a centrally controlled enterprise decision service that evaluates attributes and policy, and renders decisions that are then passed to the PEP. This allows for central management and control of subject attributes and policy. Alternatively, local organizations within the enterprise may implement separate PDPs which draw on a centralized DP store. The design and distribution of ACM components requires a management function to ensure coordination of ABAC capabilities.

To compute access decisions, the PDP must have information about the attributes. This information is provided by the PIP. The PIP is defined in this document as:

Policy Information Point (PIP): *Serves as the retrieval source of attributes, or the data required for policy evaluation to provide the information needed by the PDP to make the decisions.*

Before these policies can be enforced, they must be thoroughly tested and evaluated to ensure they meet the intended need. This action is carried out by the PAP. The PAP can be defined as:

Policy Administration Point (PAP): *Provides a user interface for creating, managing, testing, and debugging DPs and MPs, and storing these policies in the appropriate repository.*

Finally, as an optional additional component within the ACM, the Context Handler manages the order of policy and attribute retrieval. This can be important when time critical or disconnected access control decisions must be made. For example, attributes may be retrieved in advance of an access request, or cached to avoid the delay inherent in retrieval at the time of the access request. The Context Handler also coordinates with PIPs to add attribute values to the request context, and converts authorization decisions in the canonical form (e.g., XACML) to the native response format. The Context Handler can be defined as:

Context Handler: *Executes the workflow logic that defines the order in which policy and attributes are retrieved and enforced.*

3. ABAC Enterprise Considerations

Many factors must be considered before deploying an ABAC system across an enterprise. This section addresses consolidation of available guidelines based on the state of the technology to date and lessons learned through multiple attempts within the Federal Government to deploy ABAC capabilities throughout a large enterprise. The guidelines are presented according to the phases of the NIST System Development Life Cycle (SDLC) illustrated in Figure 6. For more general information regarding the definitions of the phases and expected outputs, refer to [NIST800-100]. Most considerations for employment of enterprise ABAC fall within the first four phases: Initiation, Acquisition/Development, Implementation/Assessment, and Operations/Maintenance. This section focuses on those phases exclusively.

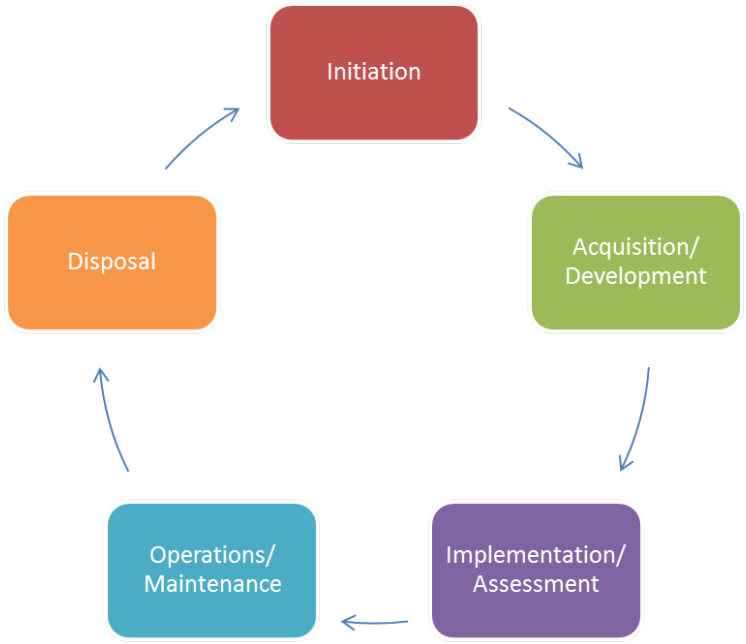


Figure 6: ACM NIST System Development Life Cycle (SDLC)

The development and deployment of an enterprise ABAC capability requires the careful consideration of a number of factors that will influence its design, security, and interoperability. These factors lead to a set of questions that should be considered:

- **Establish the Business Case for ABAC Implementation.** What are the costs of developing/acquiring new capabilities and transitioning away from old capabilities? What are the important benefits provided by ABAC? What new risks, if any, are introduced by ABAC, what new governance structures are required to manage shared capabilities and documentation of policies that were previously human-in-the-loop decisions? Which datasets, systems, applications, and networks need ABAC capabilities? How is liability for data loss or misuse of data managed?
- **Understand the Operational Requirements and Overall Enterprise Architecture.** How are privileges managed, monitored, and validated for compliance? What interfaces and objects will be exposed by the enterprise for information sharing? What ACM will be used? How will subject and object attributes be shared and managed? What are the access control rules and how are they captured, evaluated, and enforced? How is trust managed within the enterprise?
- **Establish or Refine Business Processes to Support ABAC.** Are access rules and policies fully understood and documented? How are required attributes identified and assigned? How are

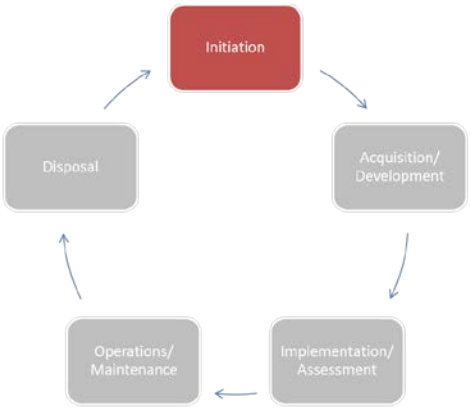
multiple policies applied in a hierarchy and deconflicted? How are access failures handled? Who creates new policies? How are common policies shared and managed?

- **Develop and Acquire an Interoperable Set of Capabilities.** How will interoperability be achieved? How are subject attributes from identity management integrated into ABAC? How are diverse or special needs for identities handled? How are subject attributes shared and maintained across enterprise entities? What are the tradeoffs with centralization versus distribution of authentication, authorization, attribute management, decision, or enforcement capability? How are environment conditions used in access decisions? How is confidence in security, quality, and accuracy measured, conveyed, and used in access decisions? How are subject attributes mapped between organizations? How are policies developed to incorporate the latest set of available subject, object, and environment condition attributes?
- **Evaluate Performance.** How are subject attributes managed for disconnected and bandwidth-limited or resource-limited users? How available are interface specifications for new participants to the enterprise? How are quality and timeliness of changes to attributes and policies measured and enforced? Will overall system and end-to-end performance be adequate?

The following sections address these principles and questions in more detail.

3.1 Initiation Phase Considerations

During the initiation phase, the organization evaluates the need for an ABAC system and its potential use. It should be determined whether the ABAC system will be an independent information system or a component of an already-defined system. Once these tasks have been completed and a need has been recognized for ABAC capabilities, several processes must take place before the ABAC system is approved, to include clearly defining goals and defining high-level requirements. During this phase, the organization defines high-level business and operational requirements as well as the enterprise architecture for the ABAC system.



3.1.1 Building the Business Case for Deploying ABAC Capabilities

As with any major system deployment, the deployment of enterprise ABAC capabilities should be preceded by significant requirements evaluation, trade studies, and planning activities to include the determination of whether ABAC is the right type of access control capability needed and feasible given the application portfolio. ABAC has the virtue of providing access without prior knowledge of or information about the subject, and large-scale enterprise information sharing of a limited set of mission or business critical objects.

Before any technical requirements are generated or deployment decisions are made, it is important to evaluate and establish a business case for the deployment of ABAC capabilities as well as to define the scope of the enterprise targeted for these capabilities. Enterprise ABAC carries with it significant development, implementation, and operations costs as well as a change in the way enterprise objects are shared and protected. Case studies or experience reports from other organizations may be helpful in planning the ABAC deployment. It may be more practical to take an incremental approach and implement ABAC protections for a limited set of objects. This implementation would establish and utilize, to the maximum extent possible, policies and attributes appropriate for the enterprise as a whole. Feedback from

incrementally building out this ABAC capability will refine policy and attribute definitions and exercise the governance and configuration management capabilities necessary to support broader ABAC use throughout the enterprise. It should be noted that without addressing the issues presented in the following subsections, an enterprise may incur significant delay and additional cost in its ABAC deployment.

3.1.2 Scalability, Feasibility, and Performance Requirements

Scalability, feasibility, and performance are important issues when considering the deployment of an ABAC product or technology. Enterprise ABAC—allowing an organization to have access to authorized objects managed by another organization in the same enterprise—requires complex interaction between ABAC components. Often these components are distributed throughout the enterprise across organization boundaries and sometimes on different networks. The larger and more diverse the enterprise, the more complex these interactions become. What may have been a simple request to access a document in a repository may now require a policy request from an enterprise service, multiple attributes from numerous logically and physically dispersed attribute sources, a third-party validation of the integrity of the object attributes bound to the document, and a decision made at one point in the enterprise while the enforcement of that decision occurs at a different point in the enterprise. Feasibility evaluation should check whether applications can support ABAC, either natively or through third-party applications. All of these potential interactions have a performance cost that must be evaluated when determining the scope of objects that may be shared through an enterprise ABAC implementation. To mitigate potential performance and scalability concerns, a variety of architectures should be considered. The distribution of ABAC components should take into account the underlying enterprise architecture, and location of necessary data and objects to be shared. For instance, PDPs and PEPs may be deployed under the same administration.

3.1.2.1 Development and Maintenance Cost

While ABAC provides many important new features when deployed across an enterprise, the cost of development, deployment, and maintenance of ABAC may exceed its benefits in the long term. In addition, the cost of retrofitting applications to use ABAC is wholly separate from procuring, setting up, and maintaining an authorization infrastructure. While cost savings can be incurred through no longer having to maintain existing solutions, it is possible that a large portion of that maintenance savings will be offset by the cost of managing and maintaining subject attributes and the policies needed for ABAC, as well as additional system support required. The benefits of having more precise[3], consistent, and flexible security must be quantified and used to determine the right balance between cost of risk and cost of security. Given these considerations, ABAC is not the right solution for every access control problem but can prove viable for environments where subjects and objects carry a rich set of attributes and access decisions involve complex relationships among these attributes.

3.1.2.2 Cost of Transition to ABAC

The governance and business process changes that must accompany the shift to ABAC represent a significant transition to an approach where objects are controlled by enterprise-governed policies and enterprise-controlled attributes, and sometimes local control as well. These objects may now need to be

[3] Attributes and rules allow more precision through a larger number of discrete inputs into an access control decision, providing a larger set of possible combinations of those variables to reflect a larger and more definitive set of possible rules, policies, or restrictions on access. ABAC allows a large number of attributes to be combined to satisfy any access control rule imaginable. As long as the attributes are available to evaluate at the time the access decision is rendered, the rule can be as complex and definitive as it needs to be to satisfy the protection requirements of the object. Thus, fine-grained AC allows access to be more detailed or flexibly partitioned when compared with coarse-grained AC, for example: coarse: employees can read file X, fine: employees working on project A can read file X, and finer: employees working on project A during office hours can read file X.

associated with an additional set of characteristics that may not have been used in access control until now. Users accustomed to logging onto their network and having broad access to resources may no longer have that luxury. While policy makers will do their best to reflect current mission and business needs in policies, there will be unexpected but inevitable denials of access to those with critical mission or business functions.

As ABAC products are implemented and an organization's access control changes, new processes and capabilities will need to be integrated into the users' day-to-day business processes and enterprise policies. During the transition it will be important to ensure that users understand why these access control changes are being implemented and what impact they will have on the way business is done. These users will need to be educated in the new ABAC systems and processes. These changes need to be properly communicated to show the benefits of an enhanced user experience, the enhanced security and safeguarding of critical information, the requirements of the new ABAC system, and the legacy access control systems, if replaced, that will be phased out. Users may be comfortable with existing processes and may not see an immediate value in switching to an ABAC capability. It may be important to emphasize areas in which ABAC enhances the security posture of the enterprise in contrast to areas where it complements existing access control mechanisms.

3.1.2.3 Need to Review Privilege and Monitor Authorizations

Some enterprises may desire the ability to review the capabilities associated with subjects and their attributes and the access control entries associated with objects and their object attributes. More succinctly, there are some requirements to know what access each individual has before the requests are made. This is sometimes referred to as "before the fact audit". Before the fact audit is often necessary to demonstrate compliance to specific regulations or directives. Another commonly desired review feature is determining who has access to a particular object or to the set of resources that are assigned to a particular object attribute. An ABAC system may not lend itself well to conducting these audits efficiently. Rather, a key feature of ABAC is the ability of the object owner to protect and share the object without any prior knowledge of individual subjects. Evaluating the set of subjects that have access to a given object requires a significant data retrieval and computation effort—possibly requiring every object owner to run a simulation of the access control request for every known subject in the enterprise. Limiting the scope of ABAC implementation can help in predetermining access authorizations, but other methods of ensuring the validity of access authorizations should be explored if the enterprise requires such validation.

3.1.2.4 Understanding Object Protection Requirements

Within the various parts of an enterprise there are a number of different operation and object types over which policy needs to be enforced. These may include operating systems, applications, data services, and database management systems. While some NLPs may exist to help determine authorized access, access to most objects is controlled through local group policy governed by local business rules, undocumented evaluation factors, and inherited non-standard doctrine. Implementing ABAC requires, first and foremost, a thorough understanding of the objects and their protection requirements. Without that understanding, the cost to develop and implement the technology required for enterprise ABAC increases dramatically. It is recommended that enterprise ABAC implementations be initially applied to objects that are well defined, controlled, and documented.

3.1.2.5 Enterprise Governance and Control

Successful enterprise ABAC requires the coordination and determination of several business process and technical factors as well as establishment of enterprise responsibilities and authorities. Without the proper governance model in place, organizations will develop stovepiped solutions and enterprise interoperability

will be delayed significantly. It is recommended that an enterprise governance body be formed to manage all identity, credential, and access management capability deployment and operation and that each subordinate organization maintain a similar body to ensure consistency in managing the deployment and transition associated with enterprise ABAC implementation. Additionally, it is recommended that the governance body develop a "trust model" that can be used to illustrate the trust chain and help determine ownership and liability of information and services, needs for additional policy and governance, and requirements for technical solutions to validate or enforce trust relationships. The trust model can be used to help influence organizations to share their information with clear expectations of how that information will be used and protected, and be able to trust the information coming from other organizations.

Additionally, enterprise authorization services should be tightly integrated with security audit, data loss prevention, security configuration management, continuous monitoring, and cyber defense capabilities. Authorization services alone are not enough to ensure the security needed to protect the mission-critical objects resident on the networks. Efforts should be undertaken to fully understand enterprise security requirements and the impacts an ABAC implementation will generate. For example, when using a distributed ACM architecture there may be consequences to the ability to audit access control decisions and events.

ABAC systems can benefit from deployment in environments governed by a trust framework. A comparison of representative trust chains for legacy ACL use and ABAC use (Figures 7 and 8) shows that there are many more complex trust relationships required for ABAC to work properly. ACLs are established by the object owner or administrator, who ultimately enforces the object access rules by provisioning access to the object through addition of a user to an ACL. In ABAC, the root of trust is derived from many sources, such as Subject Attribute Authorities or Policy Developers.

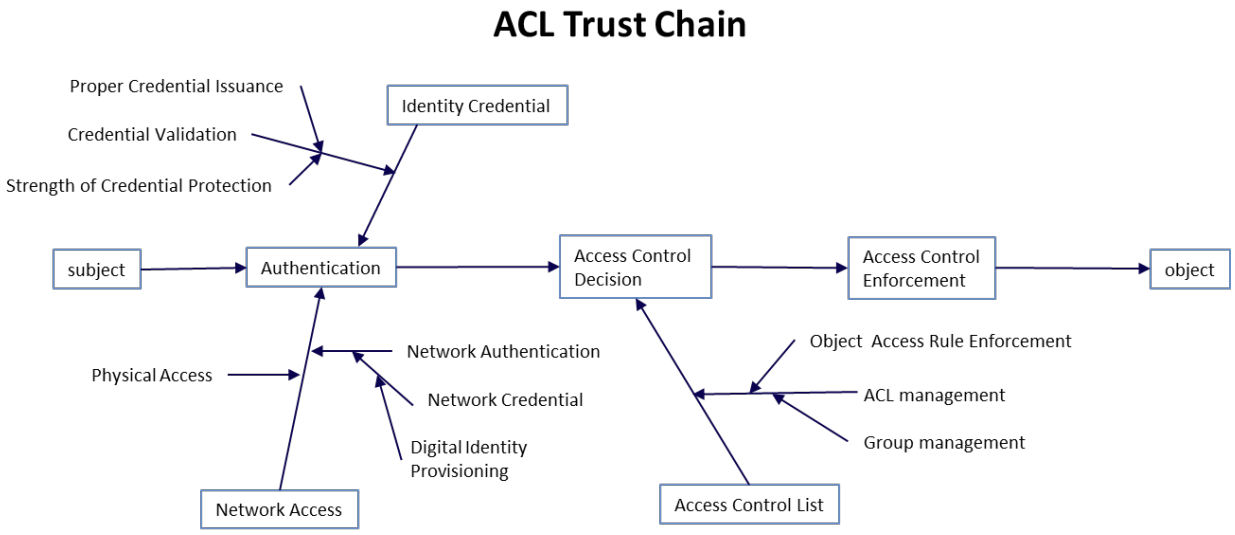


Figure 7: ACL Trust Chain

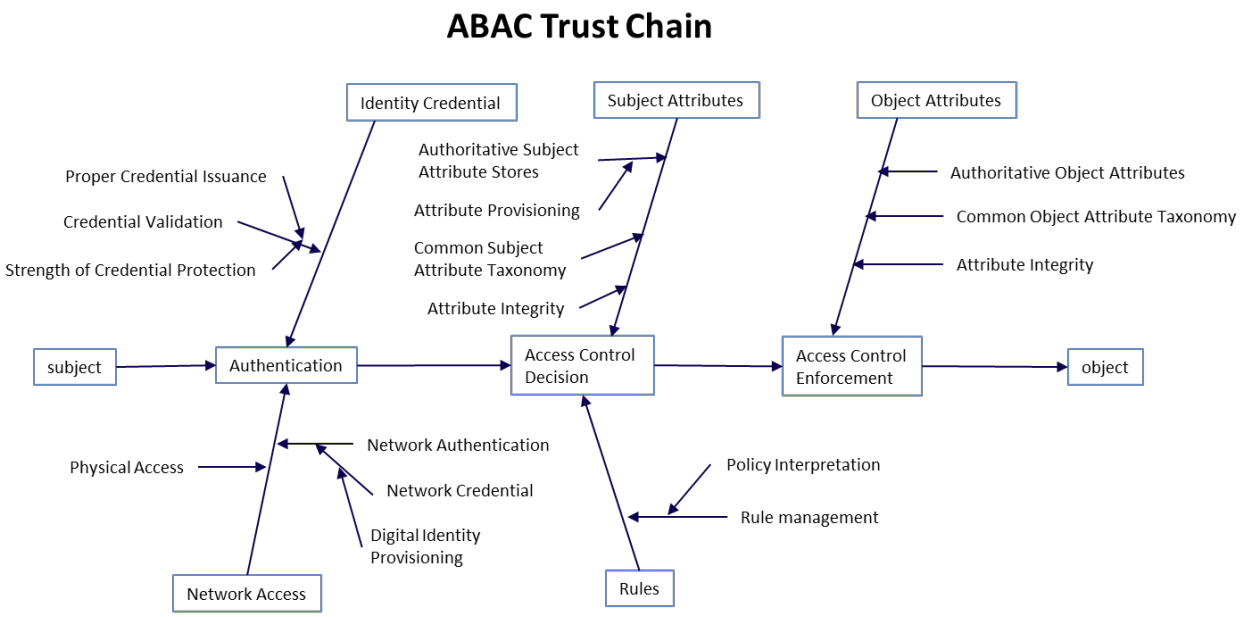


Figure 8: ABAC Trust Chain

When managing the risk inherent in information sharing, two perspectives of risk must be addressed when deploying an enterprise ABAC solution. First, an ABAC solution may be considered one of many security control options that help protect an enterprise from risk. The risk of unauthorized access to protected resources can be reduced with an ABAC implementation because precise policies can be implemented consistently and updated more easily to address changing threats. Second, use of ABAC may increase or decrease operational risk of an enterprise by exposing protected objects to access by unknown entities. By assuming that attributes are issued appropriately, the ABAC system is partially dependent upon the attribute-issuing authorities. This multiplicity of risk sources presents a number of challenges that must be managed through governance and a formal trust model.

When establishing a governance model for managing the risks inherent in ABAC, it is important to ensure there are mechanisms and agreements in place with each responsible organization to monitor and manage these roots of trust and any liabilities that occur as a result of unwarranted access.

3.1.3 Developing Operational Requirements and Architecture

Several high-level operational and architecture planning requirements must be satisfied before implementing an ABAC solution:

- First, identify the objects that will be shared and protected by ABAC.
- Second, define the rules or policies that govern their protection.
- Third, identify and define the subject and object attributes, along with their associated authorities, in coordination with the access control rule developers.
- Fourth, develop processes regarding how the access control policies are written, validated, and managed.
- Finally, determine how the ACMs will be segmented or distributed throughout the enterprise and how attribute, policy, and decision requests and responses will be rendered.

3.1.3.1 Object Identification and Policy Assignment

The objects selected to be shared and protected by the ABAC solution will vary based upon organizational requirements. Each object or class of object must be identified and the policy or rules protecting each must be documented. A set of business processes need to be established to identify, class, and assign policy to each new object created within the scope of the ABAC implementation.

3.1.3.2 Attribute Architecture

Access control policies are expressed in terms of attributes. Consequently all required attributes must be established, defined, and constrained by allowable values required by the appropriate policies. The schema for these attributes and allowable attribute values must be published to all participants to help enable object owners with rule and relationship development. Once attributes and allowable values are established, methods for provisioning attributes and appropriate attribute values to subjects and objects need to be established as well as an architecture for any attribute repositories, retrieval services, or integrity checking services. Interfaces and mechanisms must be developed or adopted to enable sharing of these attributes.

3.1.3.3 Subject Attributes

Many human subject attributes are typically provisioned upon employment with the organization and may be provisioned by several different authorities (human resources, security, organization leadership, etc.) For these, approaches to obtaining authoritative data are well known. As an example, only security authorities should be able to provision and assert clearance attributes and attribute values based on authoritative personnel clearance information; an individual should not be able to alter his or her own clearance attribute value. Other subject attributes may involve the subject's current tasking, physical location, and the device from which a request is sent; processes need to be developed to assess and assure the quality of such subject attribute data.

Authoritative subject attribute provisioning capabilities should be appropriately dependable in regards to quality, assurance, privacy, and service expectations. These expectations may be defined in an Attribute Practice Statement (APS). An APS provides a listing of the attributes that will be used throughout the enterprise, and may identify authoritative attribute sources for the enterprise. Still further network infrastructure capabilities (including the ability to maintain attribute confidentiality, integrity, and availability) are required to share and replicate authoritative subject attribute data within and across organizations.

3.1.3.4 Object Attributes

Object attributes are typically provisioned upon object creation and may be bound to the object or externally stored and referenced. It is to be expected that access control authorities cannot closely monitor all events. Frequently, this information is driven by non-security processes and requirements. Good attribute data that support good access decisions are essential, and measures must be taken to ensure that object attributes are assigned and validated by processes that the object owner or administrator considers appropriate for the application and authoritative. For example, object attributes must not be modifiable by the subject to manipulate the outcome of the access control decision. The object attributes must be made available for retrieval by access control mechanisms for access control decisions. Additional considerations for creating object attributes include:

- In general, users will not know the values of an object attribute (e.g., to which sensitive compartment a given user is authorized). This should be accounted for in ACMs, so that users only see the values that are applicable to them.
- As with subject attributes, a schema is required for object attributes defining attribute names and allowed values.
- Attributes need to be kept consistent in DP, MP, and NLP.

There have been numerous efforts within the Federal Government and commercial industry to create object attribute tagging tools that provide not only data tagging, but also cryptographic binding of the attributes to the object and validation of the object attribute fields to satisfy access control decision requirements.

3.1.3.5 Environment Condition

Environment condition refers to context information that generally is not associated with any specific subject or object but is required in the decision process. They are different from subject and object attributes in that they are not administratively created and managed, but instead are intrinsic and must be detectable by the ABAC system. Environment conditions such as the current date, time, location, threat, and system status, usually are evaluated against current matching environment variables when authorizing an access request. Environment conditions allow ABAC policies to specify exceptional or dynamic AC rules that cannot be described by subject/object attributes only. When composing ABAC rules with environment conditions, it is important to make sure that the environment condition variables and their values are globally accessible, tamper proof, and relevant for the environments where they are used.

3.1.3.6 Access Control Rules

In ABAC, all AC rules must include some combination of attributes and allowable operations. They may also include conditions, hierarchical inheritance, and complex logic. Together these provide a rich array of options when implementing ABAC. Rule sets and the application of rule sets to objects must be governed and managed appropriately. Rules must accurately and completely reflect the NLP, and be authoritatively developed (some by organizations, some by resource owners), applied, maintained, shared, and asserted. ABAC allows multiple rules from multiple stakeholders. New techniques are needed to coordinate and obtain the proper balance of sharing and protection. In some settings, one might limit the visibility of which rules apply to which objects to limit the likelihood of unauthorized subjects manipulating attributes to obtain authorization. In other circumstances, subjects that are denied access should have a method to verify or rectify the circumstances that caused the denial. Some organizations may wish to track the denials to see if the rules were appropriate. Similarly, rule definition and employment mechanisms and processes should include a robust rule deconfliction (resolution for the different decisions of rules) capability to determine rule conflicts and resolution processes.

3.1.3.7 Access Control Mechanism and Context Handling

The distribution and orchestration of ACM must be predetermined to avoid conflicts and weaknesses in object protection. For example, if an identical object is held by two different organizations, an unauthorized subject should not be able to access the version held by the organization with lesser restrictions. ACMs should be managed, maintained, and employed in a consistent manner to ensure interoperability and comprehensive security.

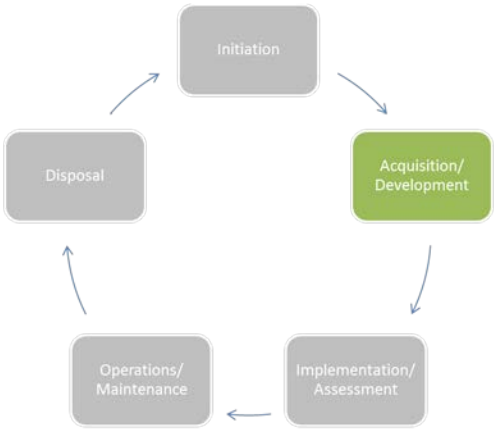
The order in which the ACM retrieves information, evaluates for a decision, and enforces the decision can differ greatly based on the specific requirements of the implementation, and may even take into account

environment conditions during access control decision rendering. This is referred to as Context Handling and simply refers to the workflow the ACM undertakes when gathering the data needed for a decision.

Additionally, where and how policy, attribute, and decision information are stored and exchanged throughout the enterprise is an important consideration, for performance and scalability purposes.

3.2 Considerations during the Acquisition/Development Phase

During the acquisition/development phase, the system is designed, purchased, programmed, developed, or otherwise constructed. Typically, during this phase, the organization prepares the business processes needed for enterprise-wide execution and defines the systems to be deployed and integrated. During the first part of this phase, the organization should simultaneously define the system's security and functional requirements. During the last part of this phase, the organization should perform developmental testing of the technical and security features/functions to ensure that they perform as intended prior to launching the implementation/assessment phase.



3.2.1 Business Process Generation and Deployment Preparation

3.2.1.1 Documentation of Rules

For each of the types of objects controlled by an organization, there should be an accompanying set of access control rules documented in an NLP. (Use cases might provide the easiest means for enterprise participants to define NLP for a set of objects.) These rules should dictate who can and cannot create, view, modify, delete, forward, and interact with data and services controlled by the organization and under what context or environment conditions they have those privileges. Documenting these rules incorporates the organization's interpretation of applicable policies and guidance, the specific sensitivities of applicable objects, and knowledge of appropriate user communities that will need the objects.

Documenting NLP facilitates the development of DP and provides traceability back to the written policy. For example, many organizations have difficulties transitioning their authorization capabilities from ACLs into a more robust ABAC infrastructure because no corresponding NLP exists. As an example, consider that when a request for access is received, the data owner evaluates a set of criteria—usually undocumented—such as, "Is this person a member of the working group?" or "Am I familiar with this person or his or her organization?" and then renders a decision before adding the requestor's name to the appropriate ACL. Well documented NLPs enable transition from human generated decision making to a consistent automated policy driven access control decision.

3.2.1.2 Customizing Policy

Unless required by higher authorities or obligations, subordinate organizations should not make local policies less stringent. If subordinate organizations in an enterprise are able to independently relax the restrictions established for enterprise policy, the security inherent in the system is undermined, possibly allowing local access to enterprise objects where it would otherwise be forbidden.

Local access policies implemented in a federated enterprise should reflect the enterprise policy associated with the requested object, based on mapped attributes from the requestor's organization. Depending on the sharing agreements between organizations, objects with shared ownership or control should be protected according to the most restrictive policy.

3.2.1.3 Agreement and Understanding of Attributes

A consistent set of valid values must be defined and applied for enterprise subject and object attributes. This allows authorization decisions to be based on known values that are consistent throughout the enterprise. The lifecycle management of attributes is the responsibility of the provisioning organization, whether the attributes are used exclusively within an organization or across organizations.

3.2.1.4 Understanding Meaning of Attributes

Attribute service providers need to describe attributes and their relationship with other attributes so that consumers may properly and effectively use attributes. Attribute service providers must document the definitions and meanings of enterprise authorization attribute values and provide guidance on the use of the attributes. In some cases, attributes must be used in combination with other attributes to establish a valid context, such as the combination of role and organization—a role has no meaning unless it is defined within the context of an organization. For example, the Director of Operations for an entire organization, whose responsibilities may encompass the Finance, Human Resources, Legal, and other departments, has an entirely different contextual meaning from the Director of Operations within the Web Services branch of the IT Department. Without the understanding of the guidance related to the attribute, its context, and the knowledge that these attribute values are required together to render a decision, the DP—and hence the decision—may be generated on insufficient information or using faulty logic.

3.2.1.5 Processes and Procedures for Access Failures

A set of procedures and requirements for communicating exception handling, access denials, and errors should be established to provide users a means to remediate access decisions given mission, role, and need-to-know imperatives. As authorization services mature from the traditional method of provisioning an account and populating an ACL to an automated decision process, it will be more difficult for system users to understand and remedy access denials. A well-established process for properly discovering and obtaining the attributes needed for access approval will help ease the transition. This can be expanded to address dropped connections or other difficulties in accessing the authorization service component.

In a mission-critical role, the subject should be able to understand the limitations and request an exception, be pointed to an authoritative source of help, or attempt an alternate path to access equivalent information or services.

3.2.1.6 Attribute Privacy Considerations

ABAC capabilities should be developed to comply with all applicable privacy laws, directives, and policy. Due to the personal and descriptive nature of subject attributes, implementing attribute sharing capabilities may increase the risk of privacy violation of personally identifiable information (PII) due to inadvertent exposure of attribute data to untrusted third parties or aggregation of sensitive information in environments less protected than the originator's. Organizations engaged in attribute sharing should employ trust agreements to ensure the proper handling of PII and enforcement of PII regulations. These trust agreements should detail authorized PII use and handling for all components in the trust chain as well as methods for validating, remediating, and adjudicating liability for regulatory infractions. A second consideration is that subject attributes can be revealed by the patterns of grant/deny decisions. If a subject

accesses a particular resource, the subject must possess attributes as specified in the access rule for that resource. The organization should protect access logs or other means of discovering grant/deny decisions.

3.2.1.7 Digital Policy Creation and Maintenance

Each DP should be specified to satisfy the requirements of an NLP. DPs are sensitive and need to be protected in the same way as objects, according to an appropriate policy. These policies may pertain to creation and modification of specific portions of the DP. DPs should be written or modified only by individuals who can interpret NLPs and have authority to write the DP. Implementing a particular NLP may require specification of multiple DPs. Special consideration should be taken to ensure that subordinate policies do not conflict with higher level policies. Individual organizations should develop and maintain local policy and unique policy that applies only to their constituent or subordinate organizations.

3.2.2 System Development and Solution Acquisition Considerations

3.2.2.1 Standardization and Interoperability within the Enterprise

Implementers of ABAC should strongly consider using a comprehensive standards-based approach that enables current day interoperability and future deployment flexibility by making use of products or capabilities that meet these objectives. An established practice to achieve interoperability and cost-efficient ABAC deployments is to use a series of standards, specifications, and standardized configurations (specifying a subset of standard options, i.e., a profile). Standards that have optional elements may be implemented inconsistently by developers, making it possible for services or applications that are fully compliant with a standard to be non-interoperable. For this reason, well-defined and standardized profiles should be encouraged, especially in cross-organizational environments. When acquiring ABAC solutions, implementers should use commonly agreed-upon tailored profiles as well as leverage the standards and profiles contained within existing standards registries.

Individual authorization service components (e.g., policy decision point, policy enforcement point, policy retrieval point, attribute retrieval point, metaattribute retrieval point) should be developed with standard, open interfaces so that systems from multiple products can be employed while ensuring interoperability. Enterprises should consider a set of requirements addressing functionality, interfaces, infrastructure, and product support to employ as a filter within the procurement process for all acquisitions regardless of categorization or affiliation.

3.2.2.2 Identity Management Integration

A request for access to an object must be authenticated as originating from a unique subject. Authentication is achieved through use of identity credentials, and must occur before an access decision can be made. The ABAC system needs to support the prevalent and strategic authentication mechanisms and credentials used by the organization. This may mean the organization needs to make enhancements to its authentication infrastructure, if its current state impedes ABAC adoption. The subject attributes conveyed in these credentials should uniquely determine the subject, and the identity vetting process used to issue credentials should be sufficient to hold the identified entity accountable. The issuance and vetting processes should be recognized throughout the enterprise as trustworthy and sufficient to enforce accountability requirements. Strong authentication methods should be used that are of sufficient assurance for the request ([NIST 800-63-1, NIST800-63-2]). Once the subject is authenticated, attributes associated with the subject can be used to determine an access decision, and access decisions can be captured in required audit records/systems to provide attribution of the request. For example, a request transferred via a Transport Layer Security (TLS) 1.2 session with client authentication [RFC5246] depending on X.509

certificates issued by a trusted certificate authority is associated to the entity bound by the certificate authority to the distinguished name.

3.2.2.3 Support for NPEs

Support for NPEs in access control services has special requirements. Authorization services use attributes associated with entities in any form. The attributes bound to the NPE not only help define the unique NPE but also reflect the context of that entity within an organization.

In some cases, an NPE subject may be acting on behalf of one or more human subjects. These NPEs may carry their own identity credentials independent of any human subject. Note that the access control system basing an access decision on an NPE credential will not be able to attribute the request to the individual or individuals who may be acting in that role, or logged into the group account, at the time of the request. NPEs may act either independently or on behalf of an authenticated individual. NPEs may include network devices (e.g., switches, routers), processes running on servers (e.g., portals), workstations, and other endpoint devices. As mission and security functions are increasingly automated, NPEs will play a larger role as actors in authorization service interactions.

3.2.2.4 Authentication and Data Integrity between ABAC Components

The authorization service requires strong mutual authentication between ABAC components (e.g., PEP, PDP) when authorization service components exchange sensitive information. For each exchange, proof of origin, data integrity, and timeliness should be considered. For example, when the authorization service needs to obtain attributes from an authoritative attribute service, mutual authentication should be used, followed by mechanisms for validating message integrity and message origin. Authentication protocols based on strong methods (e.g., X.509 authentication) should be used to provide the level of assurance needed by both parties involved in the attribute exchange.

3.2.2.5 Integrating Other Controls with ABAC

Authorization services alone are not enough to ensure the security needed to protect the mission-critical objects distributed throughout the enterprise. Comprehensive and cohesive security capabilities are needed to establish the desired level of assurance, and they must be tightly integrated and able to seamlessly feed the security information needed for making and enforcing access decisions. These other controls may include subject authentication, security audit, security configuration management, intrusion detection, and monitoring capabilities.

3.2.2.6 Selection and Accessibility of Attribute Sources

Authorities should be clearly identified so that the attribute source is able to provide attributes to the policy decision point from an authoritative source. When multiple attribute services are available, possibly with different metaattributes (such as assurance level), the attribute store/policy information point should balance the retrieval of attributes that satisfy the most restrictive policies, with performance and availability requirements.

3.2.2.7 A Shared Repository for Subject Attributes

Direct use of shared repositories for subject attributes should be considered where there is sufficient network connectivity to take advantage of economies of scale, increased quality control, and standard interfaces. Another advantage of using shared attribute repositories is that they provide a single access point for data from multiple sources. Building and managing a connection to a single access point may be

less complex than managing multiple connections. In some cases, limited connectivity, insufficient bandwidth, or intermittent connections may prevent authorization service providers from being able to use shared repositories reliably. It may be necessary to maintain local copies of data that cannot be continuously in sync with a shared attribute repository, and thus not have access to current data.

3.2.2.8 Minimum Attribute Assignments

In some enterprises, a minimum set of attributes may be defined. With a standard set of enterprise subject attributes and object attributes, DPs can more easily be developed and modified to reflect changes in policy. One example of where this approach applies is with classification and compartmentalization markings within classified networks. In most cases, an object cannot be placed on the network without proper marking, and access control policies are written to address the finite and well-known set of classification and compartmentalization markings.

3.2.2.9 Environment Conditions

When required by policy, environment (or contextual) information can be fed into the access control process. Examples include threat level, subject/object location, method of authentication, or time of day. The environment conditions may change more rapidly over time than subject and object attributes.

3.2.2.10 Attribute Management

Authorities for assigning attributes should be clearly defined and consistent with an appropriate attribute policy. Some form of validation, integrity, and provenance mechanisms (to verify the completeness, allowable values, integrity, and change history of attributes) should be integrated into the mechanism or framework used to manage attributes.

3.2.2.11 NLP/DP Traceability

A comprehensive and coherent traceability between high-level enterprise written policy/NLP and low-level enterprise or local DP should be maintained by an appropriate authority. This will enable changes to written policy to be evaluated and subsequent DPs to be altered accordingly. With this policy traceability, the plethora of DPs resident in local organizations will be auditable, verifiable, and alterable given any change to requirements.

3.2.2.12 Rules or Policies Based on the Agreed Attributes

If an organization has an agreement with one or more organizations to use a defined list of attributes (some industry and use case-specific groupings of attributes are available today), the organization that owns the objects must ensure that it writes access control policies based only on those attributes. Every effort should be made to use any accepted common set of shared enterprise attributes, no matter how limited, to ensure basic interoperability if only to effect a limited secure information sharing capability. As new requirements arise, the enterprise may choose to introduce new enterprise attributes and rules for sharing them.

For example, the OASIS XACML Export Control–US (EC-US) and Intellectual Property Control (IPC) Profiles serve as examples of domain-specific standardized attributes with generally constrained attribute values. The EC-US Profile documents the attributes common to access control decisions for the U.S. Department of Commerce Export Administration Regulations (EAR) and the U.S. Department of State International Traffic in Arms Regulations (ITAR).

3.2.2.13 Externalization of Policy Decision Services

It is common to implement PDPs as services, separate from individual enterprise services and applications. Doing so removes the burden and expense of providing similar decision services for every enterprise service or application, since a single PDP can support multiple enterprise authorization services. Allowing authorization service providers to use PDP services that are provided by the larger enterprise or by the organization greatly simplifies service/application development; saves money that would otherwise be spent on licensing, training, configuring, and deploying disparate instances of these services; and moves operations and maintenance away from individual programs.

3.2.3 Considerations for Other Enterprise ABAC Capabilities

When developing and implementing ABAC enterprise authorization capabilities, architects and program managers must keep in mind that there will inevitably be a long transition from the current access control methods in use now to the desired end state. As standards and technology mature, organizations will need to embrace concepts that enhance interoperability and promote higher assurance solutions while discarding proprietary, stovepiped solutions.

3.2.3.1 Confidence in Access Control Decisions

An access control decision is made by using the accurate, timely, and relevant data gathered from authoritative source(s) that are appropriate to the level of risk. Confidence in the access control decision depends upon timeliness, relevance, authority, and quality, reliability, and completeness of information used to compute the decision. Other factors in establishing confidence include identification and authentication processes (e.g., strength of authentication mechanism, identity vetting, credential issuance and proofing, attestation, source Internet Protocol [IP] address). When adopting a risk-based approach to ABAC, the factors discussed above should be taken into account.

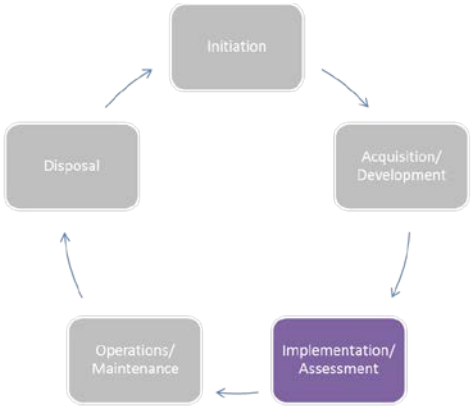
3.2.3.2 Mapping Attributes between Organizations

Organizations may name attributes and attribute values differently. It may be important to implement solutions that provide attribute mapping between enterprise organizations to minimize the need for a special class of attributes called “enterprise attributes.” Attribute mapping serves as a translation between attributes or attribute values that are named differently. For example, one organization may use the name Citizenship and another may use the name Nationality to refer to the same set of attribute values.

In practice, cross-organizational ABAC may follow a collaborative approach outlined in Sections 3.1.3.1 and 3.1.3.2. This would allow each organization to make local decisions within a framework that provides assurance of appropriate control between organizations. When new policies are created, if policy authors create or designate their own attributes, policies may not be interoperable. Using pre-agreed attributes will make the policies more uniform and easily understood.

3.3 Considerations during the Implementation/Assessment Phase

In the implementation/assessment phase, the organization installs or implements the system, configures and enables system security features, tests the functionality of these features, and finally, obtains a formal authorization to operate the system. Most of the considerations during this phase are focused on optimizing performance and ensuring security features work as expected.



3.3.1 Attribute Caching

When an ABAC solution moves from the prototype/pilot to deployment, attribute caching may be considered to enhance performance. Performance of the ABAC solution can be negatively affected if each access decision requires an across-the-network attribute request. This is especially apparent in low-bandwidth, high-latency environments.

In addition to performance issues regarding attribute caching, the organization may evaluate a tradeoff regarding the freshness of attributes and the impact upon security. Attributes that are not refreshed as often will ultimately be less secure than attributes that are refreshed in real time. For example, a subject's access privileges may have changed since the last refresh, but those updates will not be reflected in their available access privileges until the next refresh.

Environments with sporadic connectivity will need to cache attributes at the local level. The security ramifications of using cached attributes locally need to be determined within the implementing organization at a policy level, and addressed with appropriate technical controls. In these disconnected environments, administrators may employ risk-based analysis as a basis for access decisions, as some attributes at the local (disconnected) level may change or be removed before the system refreshes its attributes. The local (and disconnected system's) possible use of stale cached attributes could introduce a level of risk to the system, because the local system is not making use of the most recently available attributes. Therefore, a risk-based analysis may be warranted as to whether or not to deploy this type of solution.

An example is a deployed ship with only intermittent, non-ideal connections to enterprise network fabrics. Because the deployed user population will have only minor changes throughout their transit, supporting the "unanticipated" system user is less of a concern. In this case, a bulk download and local storage of subject attributes may be sufficient for most local access control decisions. Therefore, subject attribute data could be stored locally on the ship throughout a deployment, and local applications and services could use the data from the local store without the need to reach to an authoritative enterprise attribute source. While this is one example of a solution to an austere environment problem, it should not be inferred that this is the only solution.

3.3.2 Attribute Source Minimization

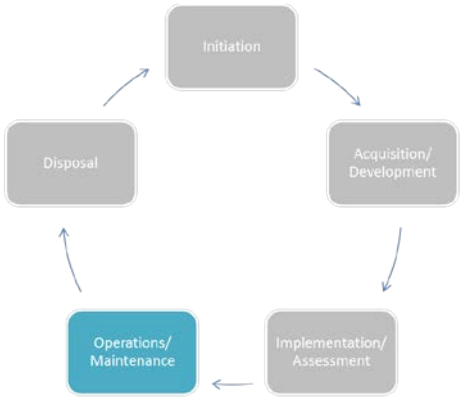
Minimizing the number of attribute sources used in authorization decisions may improve performance and simplify the overall security management of the ABAC solution. Organizations planning to deploy an ABAC solution may benefit from establishing a close working relationship among all of the organization's stakeholders who will be involved in the solution's deployment.

3.3.3 Interface Specifications

To help ensure consistently reliable access to ABAC services, all organizations that participate in information sharing through enterprise ABAC capabilities should fully understand the interface, interaction, and precondition requirements for all types of requests, including attribute and DP requests. It is also important to ensure that as changes occur in the infrastructure and interface requirements, all relying parties are provided notification of updates so they can plan to modify their components accordingly.

3.4 Considerations during the Operations/Maintenance Phase

In the operations/maintenance phase, systems and products are in place and operating, enhancements and/or modifications to the system are developed and tested, and hardware and/or software is added or replaced. During this phase, the organization should monitor performance of the system to ensure that it is consistent with preestablished user and security requirements, and needed system modifications are incorporated.



3.4.1 Availability of Quality Data

As the information needed to render access control decisions, and in some cases the decisions themselves, is externalized from the objects and consumers, access to information and services will become more dependent on an outside service's ability to provide timely and accurate data. It is important that the infrastructure be robust, well-tested, resilient, and scalable to mission needs. This is important to support attribute services, attribute stores, policy stores, policy and attribute generation and validation components, decision engines, and metaattribute repositories and conduits through which this information must pass. If outsourced, service agreements should detail availability, response time, and data quality and integrity requirements. For example, failover, redundancy, and continuity of operations must be considered for data and services that are considered mission critical. Maintaining high availability of quality data requires that addition, updating, and deleting of attribute values is performed by trained, authorized individuals, and regularly audited.

Formal agreements between providers and consumers of attributes and services should meet an appropriate standard of service, quality, availability, protection, and usage. Various laws and regulations establish responsibilities, liabilities, and penalties related to the appropriate protection of information. The agreements should capture these requirements as well as those related to responsibility for data.

Agreements establishing an appropriate level of trust between organizations are important. These agreements would serve to formalize that trust relationship with a series of requirements and, possibly, penalties for nonconformance. APSs and MOUs/MOAs for attribute services and authoritative and accountable attribute sources can also serve to translate organizational policy into operational procedures. The purpose, usage, participants, responsibilities, and administration of these services are described in these formal agreements.

4. Conclusion

This document brings together many previously separate bodies of ABAC knowledge in order to bridge existing gaps between available technology and best practice ABAC implementations and to address the emerging interest in ABAC within the Federal Government.

ABAC controls access to objects by evaluating rules against the attributes of entities (subject and object), operations, and the environment relevant to a request. ABAC relies upon the evaluation of attributes of the subject, attributes of the object, and a formal relationship or access control rule defining the allowable operations for subject-object attribute combinations. ABAC enables precise access control allowing for a large number of discrete inputs into an access control decision, providing a large set of possible combinations of those variables to reflect a diverse set of possible rules, policies, or restrictions on access. Thus, ABAC allows an unrestricted number of attributes to be combined to satisfy a rich set of policies.

This document defines general concepts necessary to understand ABAC. Specifically it addresses subject and object attributes, and generic features of an ABAC model and its components. It brings to light numerous considerations aligned to the system development lifecycle that must be factored in the planning, design, development, implementation, and operation of ABAC capabilities within an enterprise. The advantages and common pitfalls of ABAC mechanisms are discussed, especially for large enterprises.

ABAC capabilities will allow an unprecedented amount of flexibility and security while promoting information sharing between diverse organizations. It is vital that these capabilities be developed and deployed using a common foundation of concepts and functional requirements to ensure the greatest level of interoperability possible. ABAC is well suited for large enterprises. An ABAC system can implement existing role-based access control policies and can support a migration from role-based to a more granular access control policy based on many different characteristics of the individual requester. It supports the external (unexpected) user and provides more efficient administration. However, an ABAC system is more complicated, and therefore more costly to implement and maintain, than simpler access control systems.

Appendix A — Acronyms and Abbreviations

Selected acronyms and abbreviations used in the guide are defined below.

AASC	Attribute and Authorization Services Committee
ABAC	Attribute Based Access Control
AC	Access Control
ACL	Access Control List
ACM	Access Control Mechanism
APS	Attribute Practice Statement
CIO	Chief Information Officer
CONOPS	Concept of Operations
COTS	Commercial Off-the-Shelf
DAC	Discretionary Access Control
DLP	Data Loss Prevention
DoD	Department of Defense
DP	Digital Policy
DPM	Digital Policy Management
FICAM	Federal Identity, Credential, and Access Management
FISMA	Federal Information Security Management Act
GUI	Graphical User Interface
HIPAA	Health Insurance Portability and Accountability Act
IBAC	Identity Based Access Control
IETF	Internet Engineering Task Force
IP	Internet Protocol
IR	Interagency Report
IT	Information Technology
ITL	Information Technology Laboratory
MAC	Mandatory Access Control
MP	Metapolicy
NIST	National Institute of Standards and Technology
NLP	Natural Language Policy
NPE	Non-Person Entity
OASIS	Organization for the Advancement of Structured Information Standards
OMB	Office of Management and Budget
OPSEC	Operations Security
PAP	Policy Administration Point
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PII	Personally Identifiable Information
PIP	Policy Information Point
PKI	Public Key Infrastructure
RAdAC	Risk-Adaptable Access Control
RBAC	Role-Based Access Control
RFC	Request for Comment
RLS	Row Level Security
SAN	Storage Area Network
SDLC	System Development Life Cycle
SOA	Service Oriented Architecture
SP	Special Publication
SQL	Structured Query Language
TCSEC	Trusted Computer System Evaluation Criteria

TD	Technology Development
TLS	Transport Layer Security
XACML	Extensible Access Control Markup Language
XML	Extensible Markup Language

Appendix B — References

[ANSI359] InterNational Committee for Information Technology Standards, *American National Standard for Information Technology - Role Based Access Control*, ANSI/INCITS 359-2012, American National Standards Institute, New York, May 29, 2012, 56pp.

[ANSI499] InterNational Committee for Information Technology Standards, *Information technology - Next Generation Access Control – Functional Architecture (NGAC–FA)*, ANSI/INCITS 499-2013, American National Standards Institute, New York, March 19, 2013, 61pp.

[CGLO09] I. F. Cruz, R. Gjomemo, B. Lin, and M. Orsini, "A Constraint and Attribute Based Security Framework for Dynamic Role Assignment in Collaborative Environments", in *Collaborative Computing: Networking, Applications and Worksharing*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 10, 322-339 (2009). http://dx.doi.org/10.1007/978-3-642-03354-4_24.

[FEDCIO1] Federal Chief Information Officers Council, *Federal Identity, Credential, and Access Management (FICAM) Roadmap and Implementation Guidance (Version 1.0)*. Office of Management and Budget, Washington, D.C., November 10, 2009, 220pp.

[FEDCIO2] Federal Chief Information Officers Council, *Federal Identity, Credential, and Access Management (FICAM) Roadmap and Implementation Guidance (Version 2.0)*. Office of Management and Budget, Washington, D.C., December 2, 2011, 478pp. http://www.idmanagement.gov/sites/default/files/documents/FICAM_Roadmap_and_Implementation_Guidance_v2%200_20111202_0.pdf.

[FK92] D. F. Ferraiolo and D. R. Kuhn, "Role-Based Access Controls," in *Proceedings of 15th NIST-NCSC National Computer Security Conference*, National Institute of Standards and Technology, Gaithersburg, Maryland, 554-563 (1992). http://csrc.nist.gov/rbac/ferraiolo-kuhn-92.pdf.

[NIST7316] V. C. Hu, D. F. Ferraiolo, and D. R. Kuhn, *Assessment of Access Control Systems*, NISTIR 7316, National Institute of Standards and Technology, Gaithersburg, Maryland, September 2006, 60 pp. http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7316.

[NIST7657] NIST/NSA Privilege (Access) Management Workshop Collaboration Team, *A Report on the Privilege (Access) Management Workshop*, NISTIR 7657, National Institute of Standards and Technology, Gaithersburg, Maryland, March 2010, 48 pp. http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7657.

[NIST7665] *Proceedings of the Privilege Management Workshop, September 1-3, 2009*, NISTIR 7665, S. A. Durrant, T. Brewer, and A. Sokol, eds., National Institute of Standards and Technology, Gaithersburg, Maryland, January 2010, 10 pp. http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7665.

[NIST7874] V. C. Hu, and K. Scarfone, *Guidelines for Access Control System Evaluation Metrics*, NISTIR 7874, National Institute of Standards and Technology, Gaithersburg, Maryland, September 2012, 48pp. http://dx.doi.org/10.6028/NIST.IR.7874.

[NIST800-100] P. Bowen, J. Hash, and M. Wilson, *Information Security Handbook: A Guide for Managers*, NIST Special Publication 800-100, National Institute of Standards and Technology,

Gaithersburg, Maryland, October 2006 (including updates as of March 7, 2007), 178pp. http://csrc.nist.gov/publications/PubsSPs.html#SP-800-100.

[NIST800-53] Joint Task Force Transformation Initiative, *Security and Privacy Controls for Federal Information Systems and Organizations*, NIST Special Publication 800-53 Revision 4, National Institute of Standards and Technology, Gaithersburg, Maryland, April 2013(including updates as of May 7, 2013), 457pp. http://dx.doi.org/10.6028/NIST.SP.800-53r4.

[NIST800-63-1] W. E. Burr, D. F. Dodson, E. M. Newton, R. A. Perlner, W. T. Polk, S. Gupta, and E. A. Nabbus, *Electronic Authentication Guideline*, NIST Special Publication 800-63-1, National Institute of Standards and Technology, Gaithersburg, Maryland, December 2011, 121pp. http://csrc.nist.gov/publications/PubsSPs.html#SP-800-63--1.

[NIST800-63-2] W. E. Burr, D. F. Dodson, E. M. Newton, R. A. Perlner, W. T. Polk, S. Gupta, and E. A. Nabbus, *Electronic Authentication Guideline*, NIST Special Publication 800-63-2, National Institute of Standards and Technology, Gaithersburg, Maryland, August 2013, 123pp. http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-2.pdf.

[RFC5246] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol, Version 1.2*, RFC 5246, Internet Engineering Task Force, Network Working Group, August 2008, 104pp. http://www.ietf.org/rfc/rfc5246.txt.

[Sandhu et al., 1996] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models," *IEEE Computer* 29(2), 38-47 (February 1996). http://dx.doi.org/10.1109/2.485845.

[TCSEC] U.S. Department of Defense, *Information Assurance (IA)*, DoD Directive 8500.1, U.S. Department of Defense, Washington, D.C., October 24, 2002, 25 pp. http://www.prim.osd.mil/Documents/DoDD_8500_1_IA.pdf.

[WWJ04] L. Wang, D. Wijesekera, and S. Jajodia, "A Logic-based Framework for Attribute Based Access Control," in *Proceedings of the 2004 ACM workshop on Formal Methods in Security Engineering, FMSE '04*, ACM, New York (October 2004) 45-55. http://dx.doi.org/10.1145/1029133.1029140.

[XACML] "OASIS eXtensible Access Control Markup Language (XACML) TC," Organization for the Advancement of Structured Information Standards [Web page], http://www.oasis-open.org/committees/xacml/ [accessed 1/8/14].

[YT05] E. Yuan and J. Tong, "Attributed Based Access Control (ABAC) for Web Services," in *Proceedings of the 2005 IEEE International Conference on Web Services, ICWS 2005*, IEEE Computer Society, Los Alamitos, California (2005) 561 - 569. http://dx.doi.org/10.1109/ICWS.2005.25.